DOI 10.31891/2307-5732-2024-337-3-69
УДК 004.42

КРЕНЦІН МИХАЙЛО
Вінницький національний технічний університет
https://orcid.org/0000-0002-1792-9401
e-mail: mishatron98@gmail.com
КУПЕРШТЕЙН ЛЕОНІД
Вінницький національний технічний університет
https://orcid.org/0000-0001-6737-7134
e-mail: kupershtein.lm@gmail.com

МЕТОД ЗАХИЩЕНОГО ОБМІНУ ТА ЗБЕРІГАННЯ ФАЙЛІВ У ПІРИНГОВИХ МЕРЕЖАХ

У статті запропоновано метод захищеного обміну та зберігання файлів у піринговых мережах на основі протоколу IPFS (InterPlanetary File System) та регістра зсуву з лінійним зворотним зв'язком (РЗЛЗЗ). Використання IPFS забезпечує ефективне децентралізоване зберігання даних із гарантованою цілісністю завдяки застосуванню контент-орієнтованої адресації на основі геш-функцій. У IPFS кожен файл або блок даних ідентифікується унікальним CID (Content Identifier), який змінюється при будь-якій модифікації вмісту.

Для підвищення конфіденційності перед завантаженням файлів у мережу запропоновано використовувати метод шифрування на основі регістра зсуву з лінійним зворотним зв'язком (РЗЛЗЗ) та лічильників. Цей метод забезпечує ефективну генерацію псевдовипадкових перестановок байтів файлу, що унеможливлює його розшифрування без знання ключа. Шифрування із застосуванням РЗЛЗЗ характеризується високою швидкодією, простотою реалізації та низькими вимогами до обчислювальних ресурсів. Це особливо важливо для піринговых мереж, де вузли можуть мати обмежену продуктивність та нестабільне з'єднання.

Запропонований підхід усуває вразливість IPFS, пов'язану з можливістю отримання вмісту файлів на основі їхніх CID, оскільки навіть при наявності ідентифікатора доступний лише зашифрований вміст. Крім того, метод забезпечує конфіденційність переданих даних, адже файли передаються у зашифрованому вигляді і можуть бути розшифровані лише авторизованими вузлами. Метод також забезпечує високу швидкодію завдяки особливостям IPFS: файли або їх фрагменти (IPLD-об'єкти) можуть завантажуватись паралельно з кількох вузлів, що пришвидшує передачу великих обсягів даних.. Метод може застосовуватись у системах децентралізованого зберігання, платформ для захищеного обміну файлами, а також у середовищах з підвищеними вимогами до захисту інформації.

Ключові слова: пірингова мережа, геш-функції, регістр зсуву з лінійним зворотним зв'язком, шифрування, IPFS, конфіденційність.

KRENTSIN MYKHAILO, KUPERSHTEIN LEONID
Vinnytsia National Technical University

METHOD OF SECURE EXCHANGE AND STORAGE OF FILES IN HYBRID PEER-TO-PEER NETWORKS

The article proposes a method for secure file exchange and storage in peer-to-peer networks based on the IPFS (InterPlanetary File System) protocol and a linear feedback shift register (LFSR). The use of IPFS provides efficient decentralized data storage with guaranteed integrity due to the use of content-oriented addressing based on hash functions. In IPFS, each file or data block is identified by a unique CID (Content Identifier), which changes with any modification of the content. To increase confidentiality before uploading files to the network, it is proposed to use an encryption method based on a linear feedback shift register (LFSR) and counters. This method provides efficient generation of pseudo-random permutations of the file bytes, which makes it impossible to decrypt it without knowing the key. Encryption using the FLSR is characterized by high speed, ease of implementation, and low requirements for computing resources. This is especially important for peer-to-peer networks, where nodes may have limited performance and unstable connections.

The proposed approach eliminates the vulnerability of IPFS associated with the possibility of obtaining file contents based on their CIDs, since even with the presence of an identifier, only encrypted content is available. In addition, the method ensures the confidentiality of the transmitted data, since files are transmitted in encrypted form and can only be decrypted by authorized nodes.

The method also provides high performance due to the features of IPFS: files or their fragments (IPLD objects) can be downloaded in parallel from several nodes, which speeds up the transfer of large amounts of data. The method can be used in decentralized storage systems, platforms for secure file exchange, as well as in environments with increased requirements for information security.

Keywords: peer-to-peer network, hash functions, linear feedback shift register, encryption, IPFS, privacy.

Вступ

З розвитком гібридних піринговых мереж питання безпечного обміну та зберігання файлів стає все більш актуальним для корпоративних і комерційних середовищ. Гібридна архітектура поєднує централізовані та децентралізовані підходи, що дозволяє досягти високої продуктивності, масштабованості та стійкості до відмов [1]. Проте це також створює нові виклики для забезпечення конфіденційності, цілісності та доступності даних. Проблеми, пов'язані з обміном та зберіганням файлів у піринговых мережах, включають такі аспекти [2]:

1. Надійність зберігання. Через нестабільність вузлів у піринговых мережах може виникати ризик втрати даних або тимчасової недоступності файлів.
2. Пропускна здатність. Швидкість завантаження та передачі файлів часто обмежена пропускною здатністю вузлів або може знижуватися через перевантаження мережі.
3. Конфіденційність і безпека. Файли, що зберігаються в піринговій мережі, вразливі до

несанкціонованого доступу зловмисників у разі відсутності належних заходів безпеки.

4. Відмовостійкість. Випадкові або цілеспрямовані збої вузлів можуть призвести до недоступності файлів і порушення роботи мережі.

5. Управління версіями. Складність управління різними версіями файлів зростає в умовах частих змін та оновлень у децентралізованому середовищі.

6. Ідентифікація та пошук. Ефективний пошук ідентифікація файлів у великих піринговых мережах із численними вузлами та файлами є складним завданням.

7. Розподіл ресурсів. Організація ефективного розподілу ресурсів, таких як сховище даних і обчислювальна потужність, стає все складнішою із збільшенням обсягу даних і кількості учасників мережі.

Виходячи із вищеописаного, обмін та зберігання файлів у гібридних піринговых мережах вимагають ефективних рішень для забезпечення надійності, безпеки та доступності даних.

Аналіз досліджень та публікацій

Останні дослідження у сфері обміну та зберігання файлів у піринговых мережах демонструють різні підходи для підвищення ефективності та безпеки даних. Серед них особливу увагу приділяють протоколам, що забезпечують розподілене зберігання та передачу даних з використанням криптографічних методів для захисту інформації.

1. IPFS (InterPlanetary File System) – є популярним протоколом для зберігання та передачі файлів у децентралізованих мережах [3]. Цей протокол використовує гешування для ідентифікації файлів, що забезпечує їх цілісність. Кожен файл або блок даних має унікальний геш, і будь-яка зміна вмісту призведе до зміни хешу. Для підвищення конфіденційності в IPFS можуть застосовуватися зовнішні механізми шифрування перед додаванням файлів у мережу. Це дозволяє захищати дані від несанкціонованого доступу під час зберігання та передачі.

2. BitTorrent є класичним протоколом для пірингового обміну файлами [4]. Файли діляться на фрагменти, які розподіляються між вузлами. Цілісність даних забезпечується за допомогою геш-сум для кожного фрагмента, що дозволяє вузлам перевіряти отримані дані на наявність помилок або підробок. Для захисту від зловмисників можуть використовуватися додаткові шифрування на рівні передачі даних (наприклад, протокольне шифрування BitTorrent), що допомагає приховати трафік від сторонніх спостерігачів.

3. Storj – це децентралізована платформа для зберігання файлів, що використовує блокчейн [5]. Файли шифруються на стороні клієнта перед завантаженням у мережу, що забезпечує конфіденційність. Використовується метод шифрування AES-256 для захисту даних, а також механізми поділу файлів на зашифровані фрагменти, які розподіляються між вузлами мережі. Це забезпечує високу стійкість до втрати даних і ускладнює несанкціонований доступ.

4. Freenet створено для анонімного зберігання та обміну файлами [6]. У цьому протоколі дані автоматично шифруються перед додаванням у мережу, ідентифікуючи їх за допомогою ключів хешування. Freenet підтримує як тимчасове (opennet), так і постійне (darknet) зберігання файлів. Дані шифруються та розподіляються між вузлами, що забезпечує конфіденційність та захист від цензури.

5. Filecoin є блокчейн-мережею для децентралізованого зберігання даних, побудованою на основі IPFS. Безпека забезпечується шляхом криптографічного доказу збереження даних (Proof-of-Replication та Proof-of-Spacetime) [7]. Перед завантаженням дані можуть бути зашифровані користувачем, а механізми перевірки забезпечують, що вузли коректно зберігають дані протягом визначеного часу.

6. Swarm – це компонент екосистеми Ethereum для децентралізованого зберігання файлів [8]. Для захисту даних у Swarm використовуються механізми шифрування файлів та цифрові підписи для забезпечення цілісності та автентичності інформації. Дані поділяються на зашифровані шматки та зберігаються на різних вузлах мережі.

Розглянемо загальні підходи до захисту даних [9]:

1. Шифрування перед завантаженням. У багатьох піринговых системах, таких як IPFS, Storj та Freenet, користувачі шифрують файли до їх завантаження у мережу. Це забезпечує, що лише авторизовані користувачі можуть розшифрувати дані, навіть якщо вузли-учасники зберігають їх копії.

2. Хешування для контролю цілісності. Використання хеш-функцій, таких як SHA-256, забезпечує можливість перевіряти, чи не були дані змінені під час зберігання або передачі.

3. Цифрові підписи дозволяють автентифікувати відправника файлів та гарантувати, що дані не були змінені після підпису. Це підвищує довіру до джерела інформації в умовах децентралізованої мережі.

Таким чином, поєднання піринговых протоколів із сучасними криптографічними методами забезпечує надійний захист даних від втрати, модифікації та несанкціонованого доступу.

Метою роботи є: розробка методу захищеного обміну та зберігання файлів у гібридних піринговых мережах для забезпечення надійності, безпеки та доступності даних.

Виклад основного матеріалу

Для вирішення поставленої задачі пропонується взяти за основу IPFS. Він пропонує альтернативу централізованим серверам, де файли зберігаються розподілено на вузлах мережі, а доступ до них здійснюється за допомогою унікальних геш-адрес. Перевагами IPFS є [10]: розподіленість, швидкість, цілісність даних,

масштабованість та управління версіями. Недоліками IPFS є: недостатній рівень захищеності, відсутність приватності та використанні вузлів-посередників.

Для вирішення проблеми конфіденційності даних у протоколі IPFS пропонується використовувати шифрування файлів перед їх завантаженням у мережу [11]. Існує багато алгоритмів шифрування, які можна використовувати для захисту файлів (симетричні та асиметричні). Асиметричні алгоритми не є ефективними для шифрування великих обсягів даних (якими можуть бути файли). Беручи до уваги симетричні алгоритми AES, DES, 3DES, Blowfish, Twofish та їх характеристики [12], можна зробити висновок, що найкращим з них є AES, адже він має високу криптографічну стійкість, а також достатню швидкість. Проте його недоліками є фіксований розмір блоку (128, 192, 256 біт) та обмеженість у масштабуванні (не призначений для легкого масштабування на нові варіанти з більшими розмірами блоку чи ключа). Для вирішення вищеописаних недоліків пропонується перестановку байтів для шифрування файлів, а самі перестановки формувати псевдовипадковим чином. Це дозволить розбивати файл на блоки змінної довжини, а також пришвидшити процес шифрування, адже перестановка буде здійснюватися за лінійну складність. Це в свою чергу дозволить підвищити конфіденційність даних при використання протоколу IPFS для обміну файлами.

Для генерації псевдовипадкових послідовностей (ПВП) пропонується використовувати регістр зсуву з лінійним зворотним зв'язком (РЗЛЗЗ) [13]. РЗЛЗЗ являє собою регістр зсуву, в якому кожен вихідний біт визначається лінійною комбінацією попередніх бітів. Цей метод широко застосовується для створення потокових шифрів, кодування даних та у сфері телекомунікацій завдяки здатності генерувати послідовності з хорошими статистичними характеристиками та великим періодом повторення. Серед переваг РЗЛЗЗ – простота реалізації, ефективне використання ресурсів та можливість створення великих обсягів псевдовипадкових даних.

Операції перестановки часто використовуються у сучасних блокових шифрах, таких як мережа Фейстеля, SP-мережа чи алгоритм «квадрат». Однак ці архітектури виконують перестановку лише частин одного блоку або перестановку блоків у межах невеликої групи. Блоки мають фіксовану довжину, а при збільшенні розміру файлу продуктивність таких методів знижується. З огляду на ці обмеження та результати досліджень [14], пропонується комбінувати РЗЛЗЗ із лічильниками для реалізації псевдовипадкової перестановки байтів. Суть методу полягає в наступному: послідовність порядкових номерів байтів файлу розбивається на підпослідовності, для кожної з яких встановлюється лічильник зі стартовим значенням (на початку або в кінці підпослідовності). Генератор ПВП створює число, що відповідає порядковому номеру підпослідовності [15]. Вибраний байт переноситься до вихідного файлу у природному порядку (зліва направо). Після цього значення лічильника збільшується на певний крок, що визначає наступний байт для вибірки в межах відповідної підпослідовності. Процес повторюється до тих пір, поки всі байти не будуть записані у вихідний файл. У підсумку вихідний файл матиме ту ж довжину, але байти в ньому будуть переставлені у псевдовипадковому порядку. Цей метод можна розглядати як форму шифрування шляхом перестановки.

Отже, нехай є файл F, який має розмір n байт та результуючий файл (спочатку нульового розміру) F'. Операція зчитування позначається літерою R, запису W. Нехай функція отримання розміру файлу позначається як L, а тому $L(F') = 0, L(F) = n$. Для обміну файлом виконуються наступні кроки:

1. Визначається послідовність порядкових номерів байтів файлу F як P, де P_i – поточний байт файлу, $i = [0; n-1]$:

$$P = \{P_0, \dots, P_i, \dots, P_{n-1}\}, \quad (1)$$

2. Задається початковий стан регістра St_0, поліном зворотного зв'язку Sp_0. Початковий стан регістра пропонується визначати на основі ключа, який може використовуватись при обміні даними між вузлами (наприклад, на основі транспортного ключа, що генерується на основі протоколу Діффі-Геллмана для здійснення комунікації). РЗЛЗЗ пропонується використовувати довжиною в 64 біт.

3. Розбивається послідовність P на N рівних підпослідовностей (рис. 1):

$$P = \{p_0 \,||\, p_1 \,||\dots||p_k||\dots||p_{N-1}\}, \quad (2)$$

де $N \leq 32$, де N є степенем 2, $64 \; mod \; N = 0$. Виходячи зі значення N обчислюється кількість чисел q_k у кожній підпослідовності. Якщо кількість байт n не ділиться на N без остачі, то буде N підпослідовностей довжиною $q_k = \frac{n}{N}$. Якщо ж результат ділення має остачу, відмінну від нуля, то підпослідовності з 0 до $(N-2)$ складатимуться з $q_k = \left]\frac{n}{N}\right[$ чисел, а $(N-1)$-а послідовність матиме $q_k = n - \left]\frac{n}{N}\right[* (N-1)$ чисел, де] [означає округлення до більшого цілого числа. Чим більше N, тим ефективніший псевдовипадковий розподіл.

Рис. 1 – Схема розбиття послідовності чисел на 4 підпослідовності

4. Для кожної підпослідовності p_k встановлюється лічильник $C_k, k = [0; N-1]$, що слугуватиме індикатором байту, який необхідно буде зчитувати з початкового файлу F. Для кожного лічильника C_k встановлюється його початкове значення po_k, яке дорівнює початковому або кінцевому значенню у підпослідовності p_k. Пропонується визначати це псевдовипадковим чином. Розділяється послідовність біт

початкового стану регістру St_0 на m рівних підпослідовностей St_0^h по N біт, $h = [1, N], m * N = 64$ і виконується їх додавання (із відкиданням старшого розряду, що може утворитись внаслідок перенесення розрядів). В результаті отримуємо:

$$PO = (\sum_{h=1}^{N} St_0^h) \; mod \; 2^N, \tag{3}$$

де PO – отримана послідовність біт, де 0 вказує на те, що лічильник встановлюється на початкове значення блоку p_k, а 1 – на останнє значення блоку p_k. Саме значення лічильника дорівнює po_k. Таким чином, визначається крок лічильника pa_k (якщо лічильник має мінімальне значення підпослідовності, то 1, інше -1):

$$pa_k = \begin{cases} 1, \text{якщо } PO_k = 0 \\ -1, \text{якщо } PO_k = 1 \end{cases}, \tag{4}$$

5. Формується наступний стан генератора St_{i+1}:

$$St_{i+1} = CS(St_i), \tag{5}$$

де CS – функція зсуву регістра, St_i – поточний стан регістра, St_{i+1} – результат зсуву (тобто наступний стан регістра).

6. До уваги береться $log_2(N)$ молодших біт нового стану St_i, що вказують на підпослідовність p_k (тобто ці молодші біти представляють число p_k). Виконується зчитування байту P_{po_k} із файлу F та записуємо його файл F' у природньому порядку (тобто наступним зліва направо):

$$P_{po_k} = R(po_k) \tag{6}$$

$$P'_{L(F')} = W(P_{po_k}) \tag{7}$$

7. Виконується збільшення (або зменшення) значення лічильника C_k на його крок pa_k:

$$po_k = po_k + pa_k \tag{8}$$

У випадку, якщо стан лічильника досягає свого граничного значення, то лічильник помічається неактивним і наступного разу, коли генератор дасть число, що вказує на проміжок p_k операція буде пропущена і буде виконана наступна генерація для визначення підпослідовності (крок 5).

У випадку, якщо $L(F') = n - 1$, тобто залишається один байт для зчитування, то генерація нового стану РЗЛЗЗ не є необхідною і зчитується останній байт.

8. Після виконання перестановки байтів псевдовипадковим чином, утворюється файл F' довжини n, тобто $L(F') = L(F) = n$. Складність алгоритму є лінійною, тобто $O(n)$, що є досить ефективним показником. За рахунок використання РЗЛЗЗ відбувається рівномірне визначення підпослідовності p_k на кожному кроці. В результаті отримаємо функцію перестановки:

$$F' = S(F) \tag{9}$$

Функція перестановки формалізується такими параметрами:

$$S(F) = \{n, P, N, q, St, m, PO, CS, C, p, pa, po, L\} \tag{10}$$

9. Для отриманого файлу F' визначається спеціальна IPFS адреса, що представляє собою унікальний криптографічний геш-ідентифікатор CID (Content Identifier, наприклад, QmPK1s3pNYLi9ERiq3BDxKa4XosgWwFRQUydHUtz4YgpqB):

$$CID = HF(F'), \tag{11}$$

де HF – функція отримання геш-ідентифікатора. При повторному завантаженні цього ж файлу CID не змінюється, а оновленим версіям файлу надаються нові геш-ідентифікатори.

10. Файл F' додається до локального вузла IPFS. У випадку, коли розмір файлу F' більше 256Кб, то протокол IPFS автоматично розбиває його на частини по 256Кб, утворюючи так звані IPLD-об'єкти, що мають дані та посилання на частини файлу, що поєднані між собою за допомогою дерева Меркла [16]. Таким чином отримуємо функцію завантаження частини файлу в мережу:

$$LF = Add(F'), \tag{12}$$

де $Add(F')$ – функція додавання до локального вузла, LF – результат функції.

11. Відбувається реплікація файлу (або IPLD-об'єків) по іншим вузлам мережі (забезпечує доступність у випадку відключення деяких вузлів, а також швидкодію, оскільки IPLD-об'єкти можуть завантажуватись паралельно із найближчих вузлів):

$$RF = Rep(F'), \tag{13}$$

де Rep – функція реплікації, а RF – результат функції.

12. У повідомленні іншому вузлу надсилаються отриманий ідентифікатор замість самого файлу. Вузол-отримувач на основі цього ідентифікатора зможе отримати зашифрований файл (у випадку IPLD-об'єктів, вони автоматично об'єднаються у файл ще на рівні протоколу IPFS), і на основі зворотних дій псевдогенератора (вузол-отримувач має той самий транспортний ключ, що дозволяє визначити параметри St_0, PO) розшифрувати цей файл:

$$F = RFP(F'), \tag{14}$$

де RFP – функція розшифрування файлу, F – результат функції (що і дорівнює початковому файлу).

Висновки

Таким чином, розроблено метод захищеного обміну та зберігання файлів, що має наступні переваги:

1. Швидкодія. Завантаження файлу в мережу передбачає його передачу всім вузлам (або передачу всіх IPLD-об'єктів у разі поділу файлу на такі об'єкти). Отримувачеві надсилається лише CID файлу, що значно пришвидшує процес під час повторної передачі або коли файл відсутній на вузлі (наприклад, через

його видалення). Якщо вузол хоче передати файл повторно, надсилається лише його CID, який має набагато менший обсяг. Якщо файл відсутній на вузлі, він завантажується з найближчих вузлів. У разі поділу файлу на IPLD-об'єкти завантаження відбувається паралельно з кількох вузлів, що суттєво підвищує загальну швидкість завантаження.

2. Конфіденційність. Шифрування файлу за допомогою регістра зсуву з лінійним зворотним зв'язком вирішує проблему протоколу IPFS, за якою, знаючи CID, зловмисник може отримати вміст файлу. У нашому випадку навіть після отримання доступу до файлу зловмисник зможе побачити лише зашифрований вміст.

3. Цілісність даних. CID файлу є його геш-значенням, що дозволяє перевіряти цілісність файлу та виявляти зміни чи пошкодження даних.

4. Відмовостійкість. Завдяки реплікації файлів у IPFS, навіть якщо кілька вузлів вийде з ладу, файл або його IPLD-об'єкти залишатимуться доступними для завантаження з інших вузлів мережі.

Література

1. Куперштейн Л.М. Аналіз тенденцій розвитку піринговых мереж / Л.М. Куперштейн, М.Д, Кренцін // Вісник Хмельницького національного університету. – 2021. – Т. 299, № 4. – С. 26–29. DOI:10.31891/2307-5732-2021-299-4-26-29.

2. Hughes D. Monitoring Challenges and Approaches for P2P File-Sharing Systems / D. Hughes, J. Walkerdine, K. Lee // International Conference on Internet Surveillance and Protection (ICISP-06), Cote d'Azur, France. DOI:10.1109/icisp.2006.22.

3. Gnatushok S. Що таке IPFS і для чого потрібний новий протокол? [Електронний ресурс] / Sergey Gnatushok // Medium. – Режим доступу: https://medium.com/@sergey.gnatushok/що-таке-ipfs-і-для-чого-потрібний-новий-протокол-bba3d9acebd1 (дата звернення: 14.03.2024)

4. Abhinav C. BitTorrent: The Engineering behind the BitTorrent protocol [Електронний ресурс] / Abhinav C.V // Medium. – Режим доступу: https://medium.com/@abhinavcv007/bittorrent-part-1-the-engineering-behind-the-bittorrent-protocol-04e70ee01d58 (дата звернення: 10.04.2024).

5. Exploring the storj network / Sammy de Figueiredo [та ін.] // SAC '21: The 36th ACM/SIGAPP Symposium on Applied Computing, Virtual Event Republic of Korea. – New York, NY, USA, 2021. DOI:10.1145/3412841.3441908

6. What is Freenet? How Safe Is It? [Електронний ресурс] // The Cyber Express. – Режим доступу: https://thecyberexpress.com/what-is-freenet/ (дата звернення: 19.04.2024)

7. Hoogendoorn R. A Complete Guide to Filecoin: Decentralized Data Storage Explained [Електронний ресурс] / Robert Hoogendoorn // DappRadar. – Режим доступу: https://dappradar.com/blog/a-complete-guide-to-filecoin-decentralized-data-storage-explained (дата звернення: 18.04.2024).

8. Foundation S. How to upload data to the Swarm network [Електронний ресурс] / Swarm Foundation // Medium. – Режим доступу: https://medium.com/ethereum-swarm/how-to-upload-data-to-the-swarm-network-c0766c3ae381 (дата звернення: 17.04.2024)

9. Alsowail R. Secure file sharing [Електронний ресурс] : thesis / Alsowail Rakan. 2016. – Режим доступу: http://sro.sussex.ac.uk/id/eprint/63551/ (дата звернення: 19.04.2024)

10. Silicon Mechanics [Електронний ресурс] // Silicon Mechanics. – Режим доступу: https://www.siliconmechanics.com/news/5-advantages-of-interplanetary-file-system (дата звернення: 19.04.2024)

11. An open system to manage data without a central server | IPFS [Електронний ресурс] // IPFS. – Режим доступу: https://ipfs.tech/ (дата звернення: 22.04.2024)

12. DES vs 3DES vs Blowfish vs AES [Електронний ресурс] // Baeldung. – Режим доступу: https://www.baeldung.com/cs/des-vs-3des-vs-blowfish-vs-aes (дата звернення: 22.04.2024)

13. Zapfel C. An Introduction to LFSRs for Cryptography [Електронний ресурс] / Connor Zapfel // Medium. – Режим доступу: https://medium.com/@czapfel/an-introduction-to-lfsrs-for-cryptography-bf2602640e91 (дата звернення: 23.04.2024)

14. Лужецький В. А. Метод шифрування на основі перестановки блоків змінної довжини / В. А. Лужецький, І. С. Горбенко // Захист інформації. – 2015. – т. 17. № 2. – С. 169-175

15. Лужецький В. А. Метод формування перестановок довільної кількості елементів / В. А. Лужецький, І. С. Горбенко // Захист інформації. – 2013. – т. 15. № 3. – С. 262-267

16. An Introduction to IPFS [Електронний ресурс] // Infura. – Режим доступу: https://www.infura.io/blog/post/an-introduction-to-ipfs (дата звернення: 29.04.2024)

References

1. Kupershtein L. Analysis of peer-to-peer networks trends / Leonid Kupershtein, Mykhailo Krentsin // Herald of Khmelnytskyi National University. – 2021. – Vol. 299, no. 4. – P. 26–29. DOI:10.31891/2307-5732-2021-299-4-26-29

2. Hughes D. Monitoring Challenges and Approaches for P2P File-Sharing Systems / D. Hughes, J. Walkerdine, K. Lee // International Conference on Internet Surveillance and Protection (ICISP-06), Cote d'Azur, France. DOI:10.1109/icisp.2006.22

3. Gnatushok S. Що таке IPFS і для чого потрібний новий протокол? [Electronic resource] / Sergey Gnatushok // Medium. – Mode of access: https://medium.com/@sergey.gnatushok/що-таке-ipfs-і-для-чого-потрібний-новий-протокол-bba3d9acebd1 (date of access: 14.03.2024).

4. Abhinav C. BitTorrent: The Engineering behind the BitTorrent protocol [Electronic resource] / Abhinav C.V // Medium. – Mode

of access: https://medium.com/@abhinavcv007/bittorrent-part-1-the-engineering-behind-the-bittorrent-protocol-04e70ee01d58 (date of access: 10.04.2024).

5. Exploring the storj network / Sammy de Figueiredo [et al.] // SAC '21: The 36th ACM/SIGAPP Symposium on Applied Computing, Virtual Event Republic of Korea. – New York, NY, USA, 2021. DOI:10.1145/3412841.3441908

6. What is Freenet? How Safe Is It? [Electronic resource] // The Cyber Express. – Mode of access: https://thecyberexpress.com/what-is-freenet/ (date of access: 19.04.2024)

7. Hoogendoorn R. A Complete Guide to Filecoin: Decentralized Data Storage Explained [Electronic resource] / Robert Hoogendoorn // DappRadar. – Mode of access: https://dappradar.com/blog/a-complete-guide-to-filecoin-decentralized-data-storage-explained (date of access: 18.04.2024)

8. Foundation S. How to upload data to the Swarm network [Electronic resource] / Swarm Foundation // Medium. – Mode of access: https://medium.com/ethereum-swarm/how-to-upload-data-to-the-swarm-network-c0766c3ae381 (date of access: 17.04.2024)

9. Alsowail R. Secure file sharing [Electronic resource] : thesis / Alsowail Rakan, 2016. – Mode of access: http://sro.sussex.ac.uk/id/eprint/63551/ (date of access: 19.04.2024)

10. Silicon Mechanics [Electronic resource] // Silicon Mechanics. – Mode of access: https://www.siliconmechanics.com/news/5-advantages-of-interplanetary-file-system (date of access: 19.04.2024)

11. An open system to manage data without a central server | IPFS [Electronic resource] // IPFS. – Mode of access: https://ipfs.tech/ (date of access: 22.04.2024)

12. DES vs 3DES vs Blowfish vs AES [Electronic resource] // Baeldung. – Mode of access: https://www.baeldung.com/cs/des-vs-3des-vs-blowfish-vs-aes (date of access: 22.04.2024)

13. Zapfel C. An Introduction to LFSRs for Cryptography [Electronic resource] / Connor Zapfel // Medium. – Mode of access: https://medium.com/@czapfel/an-introduction-to-lfsrs-for-cryptography-bf2602640e91 (date of access: 23.04.2024)

14. Luzhetsky V. Encryption method based on permutation of variable-length blocks / V. A. Luzhetsky, I. S. Gorbenko // Information Protection. – 2015. – vol. 17. No. 2. – pp. 169-175

15. Luzhetskyi V. Method of forming permutations of an arbitrary number of elements / V. A. Luzhetskyi, I. S. Gorbenko // Information Protection. – 2013. – vol. 15. No. 3. – P. 262-267

16. An Introduction to IPFS [Electronic resource] // Infura. – Mode of access: https://www.infura.io/blog/post/an-introduction-to-ipfs (date of access: 29.04.2024)