5. Опитувальник професійної спрямованості (ОПС) Д. Голанда. *Тернопільський обласний центр зайнятості.* URL: https://ter.dcz.gov.ua/publikaciya/opytuvalnyk-profesiynoyi-spryamovanosti-ops-d-golanda (дата звернення: 20.12.2024).

УДК 004.056:004.052;004.652

Плець В. В., здобувач вищої освіти,
Потапова Н. А., канд. екон. наук., доцент,
доцент кафедри інформаційних технологій

ВИКОРИСТАННЯ ХЕШ-ФУНКЦІЙ І ХЕШ-ТАБЛИЦЬ У КІБЕРБЕЗПЕЦІ

Донецький національний університет імені Василя Стуса, м. Вінниця

Хеш-функція – це математичний алгоритм, який перетворює дані довільної довжини (текст, числа, файли тощо) у хеш-значення (також відоме як хеш або дайджест) фіксованого розміру. Незалежно від розміру вхідних даних, результатом роботи хеш-функції завжди буде рядок однакової довжини, що робить її корисною для багатьох завдань, як-от:

1. Перевірка цілісності даних: хешування використовується для створення контрольних сум, які дають змогу перевіряти, чи не було змінено дані під час передачі або зберігання.
2. Захист паролів: у базах даних замість збереження паролів зберігають їх хеші, що підвищує безпеку.
3. Цифрові підписи та сертифікати: хеш-функції використовуються в криптографії для створення підписів і перевірки автентичності даних.
4. Оптимізація пошуку: у структурах даних, як-от хеш-таблиці, хешування допомагає швидко знаходити потрібні елементи.

Хороша хеш-функція повинна відповідати таким властивостям: детермінованість (однакові вхідні дані завжди дають один і той самий хеш), швидкість (обчислення хешу має бути ефективним), стійкість до колізій (різні вхідні дані повинні давати різні хеш-значення), невідворотність (за хеш-значенням має бути практично неможливо відновити початкові дані).

Популярні хеш-функції включають MD5, SHA-1, SHA-256 та інші. Однак деякі старі функції, як-от MD5 і SHA-1, більше не вважаються безпечними через вразливість до колізій [1]. На рис. 1 та 2 наведено обчислення хешу на C# (SHA-256) та результат виконання програмного коду.

Хеш-таблиці та хеш-функції відіграють ключову роль у сучасних системах кібербезпеки завдяки своїй здатності ефективно обробляти великі обсяги даних та забезпечувати захист від несанкціонованого доступу. Хеш-функції використовуються для створення унікальних і незворотних значень, що допомагають забезпечити цілісність і автентичність даних, а також створюють основи для алгоритмів підпису та перевірки повідомлень. Натомість хеш-таблиці є важливим інструментом для швидкого пошуку та перевірки великих наборів даних, як-от списки заблокованих IP-адрес чи паролів, що підвищує ефективність кіберзахисту.

```csharp
using System;
using System.Security.Cryptography;
using System.Text;

class Program
{
    static void Main()
    {
        string data = "Example data";

        // Перетворення рядка у байти
        byte[] dataBytes = Encoding.UTF8.GetBytes(data);

        // Створення об'єкта SHA256
        using (SHA256 sha256 = SHA256.Create())
        {
            // Обчислення хешу
            byte[] hashBytes = sha256.ComputeHash(dataBytes);

            // Перетворення байтів у шістнадцятковий рядок
            StringBuilder hashString = new StringBuilder();
            foreach (byte b in hashBytes)
            {
                hashString.Append(b.ToString("x2"));
            }

            Console.WriteLine("Хеш: " + hashString);
        }
    }
}
```

Рисунок 1 – Код для обчислення хешу на C#

```
Хеш: e01757d4eae002c09e8e033373d2ea288452d6184cc3344086090972f029fa7e
```

Рисунок 2 – Результат коду

Хеш-таблиці – це структура даних для зберігання та швидкого пошуку інформації, наприклад, чорних списків IP-адрес чи інших загроз [4].

Застосування хеш-таблиць і хеш-функцій дає змогу значно знизити ризики кібератак, як-от маніпуляції з даними або спроби злому паролів. Однак важливо враховувати, що успішне використання цих технологій залежить від правильної реалізації та вибору хеш-функцій з високим рівнем безпеки, як-от SHA-256 чи інші алгоритми, що протистоять сучасним методам злому, зокрема атакам з використанням колізій.

```csharp
using System;
using System.Collections.Generic;

class Program
{
    static void Main(string[] args)
    {
        // Створення списку заблокованих IP
        HashSet<string> blacklist = new HashSet<string>
        {
            "192.168.1.1",
            "203.0.113.5",
            "10.0.0.2"
        };

        // IP-адреса для перевірки
        string ipToCheck = "203.0.113.5";

        if (IsIpBlacklisted(blacklist, ipToCheck))
        {
            Console.WriteLine($"IP {ipToCheck} заблоковано.");
        }
        else
        {
            Console.WriteLine($"IP {ipToCheck} дозволено.");
        }
    }

    static bool IsIpBlacklisted(HashSet<string> blacklist, string ip)
    {
        return blacklist.Contains(ip);
    }
}
```

Рисунок 3 – Код використання хеш-таблиці для чорного списку IP

```
IP 203.0.113.5 заблоковано.
```

Рисунок 4 – Результат коду використання хеш-таблиці для чорного списку IP

Хеш-функції застосовуються у криптографічних протоколах, як-от TLS, для забезпечення конфіденційності й автентичності [5].

```
using System;
using System.Security.Cryptography;

class Program
{
    static void Main(string[] args)
    {
        string sessionHash = GenerateSessionHash();
        Console.WriteLine("Унікальний хеш сеансу: " + sessionHash);
    }

    static string GenerateSessionHash()
    {
        using (var randomNumberGenerator = new RNGCryptoServiceProvider())
        {
            byte[] tokenBytes = new byte[32];
            randomNumberGenerator.GetBytes(tokenBytes);

            using (SHA256 sha256 = SHA256.Create())
            {
                byte[] hashBytes = sha256.ComputeHash(tokenBytes);
                return BitConverter.ToString(hashBytes).Replace("-", "").ToLower();
            }
        }
    }
}
```

Рисунок 5 – Код для генерації унікального хешу для сеансу

```
Унікальний хеш сеансу: 84c29f5f5424e9ccd5972ec34ea273146cb886efb40572d4c5fc6a4c68ae162b
```

Рисунок 6 – Результат коду для генерації унікального хешу для сеансу

Перспективними напрямами використання хеш-функцій та хеш-таблиць є: постквантова криптографія (розробка хеш-функцій, стійких до атак квантових комп’ютерів), а також блокчейн (забезпечення прозорості та незмінності транзакцій за допомогою хеш-функцій).

Отже, хеш-таблиці та хеш-функції є важливими елементами інфраструктури кібербезпеки, що забезпечують захист даних, швидкий доступ до критичних відомостей і підтримку високої надійності в умовах постійно змінюваного кіберпростору.

Список використаних джерел

1. Blog.whitebit. URL: https://blog.whitebit.com/uk/what-is-a-hash-in-blockchain/
2. Thekernel.ua. URL: https://thekernel.ua/identyfikatsiia-avtentyfikatsiia-ta-avtoryzatsiia/
3. support.microsoft.com. URL: https://support.microsoft.com/ukua/office/%D0%B4%D0%BE%D0%B4%D0%B0%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F-%D0%B0%D0%B1%D0%BE
4. Vpnunlimited.com. URL: https://www.vpnunlimited.com/ua/help/cybersecurity/hashing?srsltid= AfmBOopWKyHmylsoS3Bf0VueGMLDPIIIT8oKC-4ZUTvOHq9gwXs2UB6C
5. Vpnunlimited.com. URL: https://www.vpnunlimited.com/ua/help/cybersecurity/hashing?srsltid=AfmBOopWKyHmylsoS3Bf0VueGMLDPIIIT8oKC-4ZUTvOHq9gwXs2UB6C