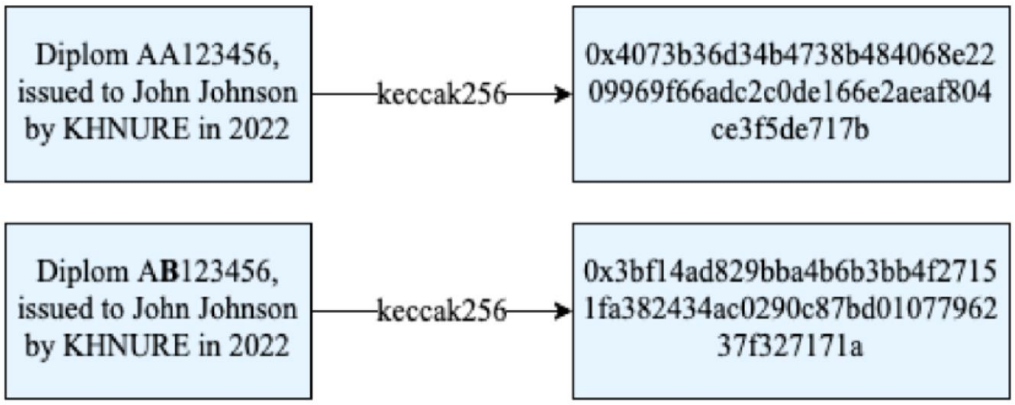


Рисунок 1 – Приклад хешування даних методом keccak256

Враховуючи безпеку та низький рівень можливості атак на SHA-3, цей алгоритм знаходить все більше застосування. Але на цьому все не закінчується. У майбутньому будуть розроблені нові та більш інноваційні хеш-алгоритми. Для криптографічної індустрії важливо постійно еволюціонувати та вирішувати численні проблеми.

Список використаних джерел

1. Antonopoulos A. M., Wood G. Mastering Ethereum: building smart contracts and DApps. Sebastopol: O'Reilly Media, 2018. 424 p.

2. Fast software encryption: 19th International Workshop, FSE 2012, Washington, DC, USA, March 19–21. 2012: revised selected papers / ed. by A. Canteaut. Berlin; Heidelberg: Springer. 2012. 476 p.

3. Group I. S. Trends in the development of science in the modern world. Primedia eLaunch LLC. 2022.

УДК 004.422.5

Головня С. О., здобувачка 2 курсу спеціальності 122 Комп'ютерні науки,
Павлюк О. А., канд. техн. наук, старший викладач кафедри інформаційних технологій

МАШИНА ТЮРІНГА ЯК МОДЕЛЬ ОБЧИСЛЕНЬ: ТЕОРЕТИЧНІ ОСНОВИ ТА СУЧАСНІ ІНТЕРПРЕТАЦІЇ

Донецький національний університет імені Василя Стуса, м. Вінниця

Проблема формального визначення алгоритму є центральною в розвитку теорії обчислень. Щоб описати процес обчислення у строгих математичних термінах, потрібна універсальна модель, здатна відтворювати будь-які логічні та арифметичні дії. Таку модель у 1936 р. запропонував Алан Тюрінг, створивши

абстрактну машину Тюрінга. У своїй праці «On Computable Numbers…» він визначив поняття обчислюваних чисел і довів, що не існує універсального алгоритму, який може встановити істинність будь-якого математичного твердження, тобто проблема розв'язності (Entscheidungsproblem) не має загального рішення [1]. Запропонована модель стала підґрунтям тези Черча–Тюрінга, згідно з якою будь-який ефективний алгоритм може бути реалізований у межах машини Тюрінга [1], [2]. Ця ідея визначила сучасне розуміння процесу обчислень і меж можливого в алгоритмах.

Машина Тюрінга – це абстрактний автомат, що складається з нескінченної стрічки, головки читання-запису та пристрою керування зі скінченною кількістю станів. Під час роботи машина читає символ, змінює його відповідно до таблиці переходів, переміщується ліворуч або праворуч і переходить у новий стан. Такий простий механізм, попри свою абстрактність, може відтворювати будь-який обчислювальний процес [1].

Тюрінг довів існування необчислюваних задач, зокрема проблеми зупинки – неможливо створити алгоритм, який визначає, чи завершиться робота будь-якої програми для заданих вхідних даних [1]. Цей результат визначив межі обчислюваності й показав, що не всі задачі мають алгоритмічне розв'язання.

Водночас учений показав, що всі обчислювані функції можуть бути реалізовані на машині Тюрінга, а універсальна машина здатна імітувати роботу будь-якої іншої [2], [3]. Це стало основою для створення концепції універсального комп'ютера, який виконує будь-яку програму, задану послідовністю інструкцій [3].

У сучасній інформатиці машина Тюрінга залишається базовою моделлю. Вона використовується для аналізу складності обчислень, формалізації алгоритмів і класифікації задач за складністю (класи P, NP тощо) [2]. В освітній практиці ця модель допомагає студентам зрозуміти логіку побудови алгоритмів і межі їх можливостей [3].

Машина Тюрінга є фундаментом сучасної теорії обчислень. Вона формалізувала поняття алгоритму, дала змогу визначити межі розв'язності задач і стала концептуальною основою для створення універсальних комп'ютерів. Її універсальність зберігає значення як у теоретичних дослідженнях, так і в освіті, забезпечуючи розуміння принципів і меж обчислювальних процесів.

Список використаних джерел

1. Тюрінг А. М. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society.* 1937. Vol. 42. P. 230–265.

2. Новаторський М. А. Алгоритми та методи обчислень. Київ: КПІ ім. Ігоря Сікорського, 2019. 256 с.

3. Ящик О. Б. Машина Тюрінга як універсальний виконавець алгоритмів та її застосування в процесі поглибленого вивчення алгоритмізації і основ програмування. *Інформаційні технології і засоби навчання.* 2016. Т. 52, № 2. С. 10–16.