**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP THỰC PHẨM TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI BÁO CÁO ĐỀ TÀI**

**NGHIÊN CỨU VÀ XÂY DỰNG MẠNG BLOCKCHAIN BẰNG NGÔN NGỮ PYTHON**

Môn học: Xử lý và tính toán song song

Giảng viên: Nguyễn Văn Tùng

Nhóm thực hiện: Nhóm 5

*TP HCM, ngày tháng năm*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DANH SÁCH NHÓM** | | | | |
| **STT** | **Họ và tên** | **Mã số sinh viên** | **Đánh giá mức độ hoàn thành (%)** | **Ký tên xác nhận** |
| 1.  Nhóm trưởng | Nguyễn Chí Thanh (NT) | 2001170170 |  |  |
| 2. | Nguyễn Nhật Trường | 2001190907 |  |  |
| 3. | Phùng Minh Khoa | 2001190618 |  |  |
| 4. | Nguyễn Hải Anh | 2001181017 |  |  |
| 5. | Phạm Kim Hoàng Hiến | 2001190523 |  |  |

|  |  |  |
| --- | --- | --- |
| **PHÂN CÔNG CÔNG VIỆC TRONG NHÓM** | | |
| **STT** | **Họ & tên** | **Nội dung công việc thực hiện** |
| 1.  Nhóm trưởng | Nguyễn Chí Thanh (NT) | Tìm hiểu về mã hóa và hash dữ liệu |
| 2. | Nguyễn Nhật Trường | Tìm hiểu về ngôn ngữ python, hàm băm , SHA-256 , thuật toán Proof of work , code demo ví dụ cho hàm băm , SHA-256 ,POW |
| 3. | Phùng Minh Khoa | tìm hiểu kiến trúc cơ bản và cơ chế hoạt động blockchain, tìm hiểu cấu trúc dữ liệu cần lưu trữ trong các block, tìm hiểu cấu trúc chuỗi khối, cơ chế liên kết các khối, nối các khối thành chuỗi và tìm hiểu hàm băm |
| 4. | Nguyễn Hải Anh |  |
| 5. | Phạm Kim Hoàng Hiến |  |

**MỤC LỤC**

[Phần 1 TÌM HIỂU NGÔN NGỮ LẬP TRÌNH PYTHON 1](#_Toc85400137)

[1.1 Khái Niệm 1](#_Toc85400138)

[1.2 Đặc Điểm 1](#_Toc85400139)

[1.3 Các ứng dụng của Python 2](#_Toc85400140)

[1.3.1 Thư viện (libraries) trong Python là gì? 3](#_Toc85400141)

[1.3.2 Thư viện python theo từng ngành nghề: 3](#_Toc85400142)

[**DEMO THƯ VIỆN numpy** 6](#_Toc85400143)

[**Các thư viện trong DEMO Blockchain(DATETIE,HASHLIB)** 6](#_Toc85400144)

[1.4 Các phiên bản Python đã phát hành 6](#_Toc85400145)

[Phần 2 CÁC FRAMEWORK THƯỜNG ĐƯỢC DÙNG TRONG LẬP TRÌNH PYTHON 8](#_Toc85400146)

[2.1 Các kiểu dữ liệu cơ bản trong Python: 8](#_Toc85400147)

[2.2 Khai báo biến trong Python 9](#_Toc85400148)

[Phần 3 CÔNG NGHỆ BLOCKCHAIN 10](#_Toc85400149)

[3.1 Khái niện blockchain 10](#_Toc85400150)

[3.2 Mã hóa 13](#_Toc85400151)

[3.2.1 Chức năng chính của mã hóa dữ liệu 13](#_Toc85400152)

[3.2.2 Các phương pháp mã hóa dữ liệu phổ biến 13](#_Toc85400153)

[3.3 Hash (hàm băm) là gì và cách thức hoạt động như thế nào? 13](#_Toc85400154)

[3.3.1 Hash là gì? 14](#_Toc85400155)

[3.3.2 Ứng dụng của Hash 15](#_Toc85400156)

[3.4 Thuật toán SHA-256 là gì? 17](#_Toc85400157)

[3.4.1 Mã hoá SHA là gì? 17](#_Toc85400158)

[3.4.2 SHA-256 là gì? 18](#_Toc85400159)

[3.4.3 Ứng dụng của SHA-256 19](#_Toc85400160)

[3.4.4 Liệu có thể tấn công Brute Force vào SHA-256 không? 19](#_Toc85400161)

[3.5 PROOF OF WORK 20](#_Toc85400162)

[3.5.1 POW là gì? 20](#_Toc85400163)

[3.5.2 Nguyên lí hđ 20](#_Toc85400164)

[3.5.3 Đánh giá ưu nhược điểm của Proof of Work 21](#_Toc85400165)

[3.5.4 Tầm quan trọng của Proof of Work 22](#_Toc85400166)

[3.5.5 Kết luận 22](#_Toc85400167)

# TÌM HIỂU NGÔN NGỮ LẬP TRÌNH PYTHON

## Khái Niệm

Python là ngôn ngữ lập trình hướng đối tượng đa năng có cấu trúc dữ liệu cấp cao mạnh mẽ và hệ thống thư viện lớn. Python hoàn toàn tạo kiểu động và sử dụng cơ chế cấp phát bộ nhớ tự động, cú pháp rõ ràng, đơn giản. Lợi ích khi học lập trình Python là gì? Python giúp người học dễ tiếp cận, bao gồm những người chưa từng học lập trình



*Nguồn:* Internet

* **Giới thiệu IDE phổ biến trong lập trình Python như sau:**
* PyCharm
* Spyder
* PyDev
* Atom
* Wing
* Jupyter Notebook
* Thonny
* Rodeo
* Microsoft Visual Studio

## Đặc Điểm

**Ngôn ngữ lập trình nhập môn:** Rossum hướng đến sự đơn giản và dễ hiểu nên cấu trúc Python khá rõ ràng, cho phép viết mã lệnh với số lần gõ phím tối thiểu. Ngày nay, Python được đánh giá là ngôn ngữ lý tưởng cho việc bắt đầu học lập trình.

**Mã nguồn mở và hoàn toàn miễn phí:** Khác với nhiều ngôn ngữ lập trình bậc cao, Python cho phép sử dụng một cách miễn phí tất cả phần mềm, chương trình được viết. Mã nguồn mở sở hữu cộng đồng đông đảo, giúp bạn giải đáp mọi thắc mắc và liên tục cập nhật, cải thiện.

**Tương thích nhiều nền tảng:** Python dễ dàng tương thích với nhiều nền tảng như Windows, MacOS, Linux. Thậm chí là di chuyển qua lại giữa các nền tảng cũng không gặp nhiều khó khăn.

**Khả năng nhúng và mở rộng:** Có thể kết hợp cùng các ngôn ngữ lập trình khác để phát triển các ứng dụng phức tạp.

**Tự động chuyển đổi code:** Khi dùng Python, người sử dụng hoàn toàn không phải lo các vấn đề về quản lý bộ nhớ, dọn dẹp dữ liệu… Python sẽ tự động chuyển code sang ngôn ngữ máy tính có thể hiểu.

**Thư viện lớn:** Python có sẵn một lượng lớn các thư viện có thể được import vào bất kỳ trường hợp nào, hay được sử dụng trong một chương trình cụ thể. Các thư viện này cho phép bạn import những đoạn code đã tồn tại trong thư viện nếu phù hợp, thay vì tự viết tất cả từ A đến Z.

VD: Trong lập trình song song:

Import threading

**Ngôn ngữ lập trình hướng đối tượng:** Dù được đơn giản hóa cho người mới học song Python vẫn giữ bản chất hướng đối tượng. Giải quyết các vấn đề trong lập trình một cách trực quan, dễ hiểu.

## Các ứng dụng của Python

**Lập trình ứng dụng web**

Ứng dụng đầu tiên có thể sử dụng ngôn ngữ lập trình Python để viết là các ứng dụng web. Ví dụ các ứng dụng web như Django, Flask, Pyramid, Plone, Django CMS. Các trang như Mozilla, Reddit, Instagram và PBS đều được viết bằng Python.

**Khoa học và tính toán**

Trong Python có chứa rất nhiều thư viện khoa học và tính toán để phục vụ cho mục đích trong tính toán. Ví dụ một số thư viện về khoa học cụ thể như EarthPy, AstroPy,…

**Tạo nguyên mẫu phần mềm**

Python có thể sử dụng để tạo ra các bản chạy thử một cách nhanh nhất. Ví dụ như các ứng dụng game. Sau khi tạo ra những bản chạy thử thì bạn có thể sử dụng các ngôn ngữ lập trình khác để viết ứng dụng game thực sự.

**Ngôn ngữ tốt để dạy lập trình**

Hầu hết các trung tâm, công ty, trường học,… đều sử dụng Python để dạy lập trình cho người mới bắt đầu [**khóa học lập trình**](https://mindx.edu.vn/course/code-for-everyone).

### Thư viện (libraries) trong Python là gì?

Đầu tiên, bạn nên hiểu rằng các thư viện Python không khác với các thư viện thông thường chứa những cuốn sách. Cả hai đều là bộ sưu tập các nguồn thông tin.

Tuy nhiên, thay vì sách, bạn sẽ truy xuất các mô-đun áp dụng trong quá trình coding của mình. Tất cả các nhà phát triển chuyên nghiệp tận dụng các mô-đun tài liệu thường được ghi lại cho quá trình học Python của mình.

Trong quá trình học Python và khi bắt đầu nghiên cứu các thư viện Python, bạn sẽ bị chìm đắm bởi số lượng thư viện chính và bên thứ ba. Có rất nhiều bộ sưu tập các mô-đun có sẵn. Bạn có thể cảm thấy băn khoăn khi không biết phải quyết định khám phá cái nào trong số chúng.

Python là một ngôn ngữ linh hoạt. Nó là một viên ngọc quý trong thế giới lập trình. Vì cách sử dụng của nó sẽ được sử dụng từ khoa học dữ liệu, phát triển web và thậm chí đến cả học máy.

các thư viện Python khác nhau sẽ bao gồm các mô-đun cho các lĩnh vực cụ thể. Dien Hinh lacác thư viện TensorFlow, PyTorch, Numpy, Sklearn

### Thư viện python theo từng ngành nghề:

1. Thư viện python cho ngành khoa học dữ liệu và thuật toán:

* Numpy– Xử lý mảng đa chiều, ma tran: *NumPy là gì?*

Bạn sẽ có thể hiểu mục đích chung của thư viện này sau khi biết tên đầy đủ của nó: Numerical Python (Python số). Nó có nghĩa là các mô-đun **xử lý số**. NumPy là phần mềm mã nguồn mở để tạo và quản lý các **mảng** và **ma trận đa chiều**. Thư viện này bao gồm một loạt các chức năng để xử lý các mảng phức tạp như vậy.

Vậy, NumPy là gì? Đây là một trong những thư viện Python, chuyên cung cấp các hàm toán học cấp cao để quản lý các mảng đa chiều. Bằng cách truy xuất các mô-đun từ NumPy, bạn sẽ hoàn thành các **tính toán** đúng đắn và chính xác. Hơn thế nữa bạn sẽ cải thiện đáng kể việc sử dụng Python với các cấu trúc dữ liệu này.

* Sklearn – Thuật toán và công cụ toán học
* Matplotlib – Vẽ đồ thị 2D :

Để thực hiện các suy luận thống kê cần thiết, cần phải trực quan hóa dữ liệu của bạn và Matplotlib là một trong những giải pháp như vậy cho người dùng Python. Nó là một thư viện vẽ đồ thị rất mạnh mẽ hữu ích cho những người làm việc với Python và NumPy. Module được sử dụng nhiều nhất của Matplotib là Pyplot cung cấp giao diện như MATLAB nhưng thay vào đó, nó sử dụng Python và nó là nguồn mở.

1. Thư viện Sklearn: mục đích sử dụng

Ví dụ cuối cùng của các thư viện Python là Sklearn, được phát triển vào năm 2007. Tuy là cuối cùng nhưng không kém phần quan trọng, vì nó cũng được đánh giá cao bởi các nhà phát triển làm việc với học máy. Sklearn (còn được gọi là scikit-learn) là một thư viện, để nhóm một nhóm các đối tượng không được gắn nhãn, ước tính mối quan hệ giữa các biến và xác định phân loại các quan sát mới.

Nói cách khác, có thể lấy một số lượng lớn các thuật toán học tập để học máy hiệu quả hơn. Thư viện Python miễn phí - Sklearn là một công cụ rất hữu ích cho mô hình thống kê và, tất nhiên, học máy!

1. Thư viện python cho máy học và học sâu, trí tuệ nhân tạo vd:

* Keras – Deep Learning
* Keras là một thư viện dựa trên Python để phát triển các mô hình học tập sâu.
* Nó cũng tương thích với các thư viện Python khác (TensorFlow hoặc Theano).
* Mục đích chính của khung framework này là tạo mẫu nhanh của các mạng nơ-ron.
* Các nhà phát triển có thể thử nghiệm với các mạng nơ-ron sâu và đào tạo chúng
* TensorFlow – Machine Learning
* TensorFlow là một thư viện mã nguồn mở.
* Tính toán số trong mô-đun này tích hợp việc sử dụng biểu đồ luồng dữ liệu.
* Đưa ra một viễn cảnh khác về khái niệm hóa các hành động toán học.
* Với khung framework này, bạn có thể tạo các mạng nơ-ron và đào tạo chúng.
* Theano – Trình biên dịch tối ưu hóa
* Theano là một thư viện dựa trên Python để thực hiện các hoạt động toán học cho các mảng đa chiều.
* Khung framework sử dụng GPU thay vì CPU, dẫn đến mức năng suất cao hơn.
* Với Theano, các nhà phát triển tạo ra các mô hình học tập sâu hoặc thư viện wrapper library.
* Một lợi thế khác là khả năng quản lý các loại tính toán cần thiết cho các thuật toán mạng nơ-ron lớn.

1. Thư viện python cho lĩnh vực lập trình web/app

* *Requests – Thư viện HTTP*
* *Scrapy- trích xuất dữ liệu từ API*

**API** (application programming interface) là tên viết tắt của giao diện lập trình ứng dụng. Nó mở ra cửa sổ cho các tương tác giữa các ứng dụng thông qua giao tiếp giữa máy với máy. Python có các khung framework làm tăng tốc quá trình tạo API. Do đó, nhiệm vụ của chúng tôi là thảo luận ngắn gọn về các thư viện Python phổ biến nhất có thể cho trong quá trình làm việc và học lập trình Python:

* *Flask,Django- Thiết kế API – Endpoint*

**Flask** là một framework trang web phát triển nhanh, được tạo ra cho quy trình thiết kế API hiệu quả hơn. Đây chỉ là một trong những cách sử dụng của Flask.

Nói chung, nó là một khung framework để phát triển ứng dụng web.

Flash rất nhẹ, cung cấp hỗ trợ để kiểm tra đơn vị và cookie an toàn cho các phiên phía khách hàng.

Các nhà phát triển khen rằng khung framework này là tài liệu tốt, có nghĩa là bạn sẽ tìm thấy nhiều trường hợp sử dụng để tìm hiểu.

**Django**

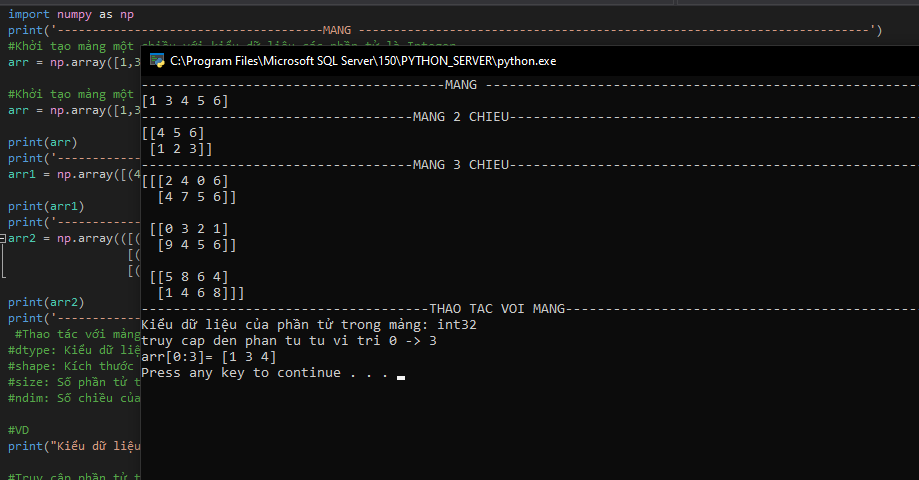
Django là một khung framework trang web bên thứ ba dựa trên Python khác.

Trong số các thư viện Python khác, mục đích chính của Djangos trong khung framework này là đơn giản hóa quá trình phát triển các trang web dựa trên cơ sở dữ liệu phức tạp.

Thư viện Django cung cấp nhiều công cụ quản lý. Do đó, các nhà phát triển sẽ có thể tạo ra các phần mã code mà không phải chuyển sang các công cụ khác.

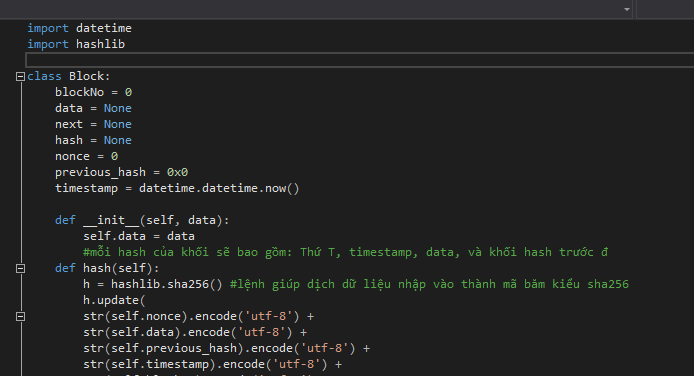
Django REST là khung framework để tạo các Web API với mã code tối thiểu.

### **DEMO THƯ VIỆN numpy**



### **Các thư viện trong DEMO Blockchain(DATETIE,HASHLIB)**

**\*import**



## Các phiên bản Python đã phát hành

Tính cho tới thời điểm hiện đại thì Python đã có 6 phiên bản được phát hành. Phiên bản đầu tiên được ra đời vào tháng 1/1994 và phiên bản mới đây nhất là vào ngày 23/12/20016. Những phiên bản của Python đó là:

Python 1.0: Đây là phiên bản đầu tiên được phát hành vào 01/1994.

Python 1.6: Phiên bản 1.x cuối cùng phát hành vào 05/09/2000.

Python 2.0: Phiên bản giới thiệu list comprehension vào 16/10/2000.

Python 2.7: Phiên bản 2.x cuối cùng phát hành vào 03/07/2010.

Python 3.0: Phiên bản loại bỏ cấu trúc và mô đun trùng lặp phát hành vào 3/12/2008.

Python 3.6: Phiên bản mới nhất tính cho tới thời điểm hiện tại phát hành vào 23/12//2016

# CÁC FRAMEWORK THƯỜNG ĐƯỢC DÙNG TRONG LẬP TRÌNH PYTHON

**Django** và **Flask** là hai web framework nổi tiếng nhất trong python

**Django** được thiết kể để nhà phát triển có thể phát triển ứng dụng web của mình một cách nhanh chóng và sạch sẽ. Django cung cấp đủ features cho phát triển ứng dụng web như template, form, routing, authentication, basic database administration, and more ...Vì vậy, Django là sự lựa chọn tốt cho những ai đã biết Python trước đó và muốn phát triển ứng dụng một cách nhanh chóng nhất.

**Flask** là microframework nhỏ và nhẹ, phù hợp cho người mới bắt đầu. . Có thể build một web đơn giản chỉ với 7 lines code.

**Jupyter Notebooks** thì phổ biến trong giới khoa học, những người nghiên cứu Học Máy (Machine Learning) và những người theo khía cạnh hàn lâm hơn của máy tính nói chung. Chúng cho phép bạn chia sẻ các đoạn code có thể chạy được cùng với giải thích về những gì code đang thực hiện.

## Các kiểu dữ liệu cơ bản trong Python:

Kiểu **int**: Kiểu số nguyên (không có chứa dấu chấm thập phân), có thể lưu các số nguyên âm và dương.

Ví dụ: 113, -114

Kiểu **float**: Kiểu số thực (có chứa dấu chấm thập phân),

ví dụ: 5.2, -7.3

Kiểu **complex:** Kiểu số phức,

**ví dụ 1:** z = 2+3j thì 2 là phần thực, 3 là phần ảo (j là từ khóa để đánh dấu phần ảo)

**ví dụ 2:** z=complex(2,3) thì 2 là phần thực, 3 là phần ảo

khi xuất kết quả ta có thể xuất:

print(“Phần thực= “,z.real) ==>Phần thực= 2

print(“Phần ảo= “,z.imag) ==> Phần ảo= 3

Kiểu str: Kiểu chuỗi, để trong nháy đôi hoặc nháy đơn

Ví dụ: “Obama”, ‘Putin’

**Kiểu bool:** Kiểu luận lý, để lưu True hoặc False

Ví dụ 1: t1=True

Ví dụ 2: t2=False

## Khai báo biến trong Python

Trong Python một biến không cần khai báo kiểu dữ liệu, khi ta gán giá trị thì tự động Python sẽ nội suy ra kiểu dữ liệu của biến. Như vậy một biến có thể có nhiều kiểu dữ liệu tùy thuộc vào giá trị mà ta gán. Ta có thể dùng hàm **type()** để kiểm tra kiểu dữ liệu của biến

# CÔNG NGHỆ BLOCKCHAIN

## Khái niện blockchain

* Blockchain là công nghệ ghi chép thông tin.
* Các khối thông tin này được lưu trữ trên các máy chủ không cố định trên toàn thế giới nên rất khó có thể đánh cắp cũng như thay đổi thông tin sai lệch vào.
* Nó được lập trình để có thể ghi chép và theo dõi bất cứ thứ gì:
* Giao dịch tài chính
* Hồ sơ y tế
* Sở hữu đất đai và nhiều thứ khác
* *Ứng dụng trong chuỗi cung ứng dược phẩm*: với Blockchain, chuỗi cung ứng dược phẩm được giám sát và truy vết rất rõ ràng, truy xuất nguồn gốc dược phẩm giúp người tiêu thụ có thể tránh dược những sản phẩm giả, kém chất lượng
* NGOÀI BITCOIN RA CÒN ỨNG DỤNG NÀO KHÁC KHÔNG

Blockchain gồm 3 hệ thống:

**Public:** Bất kỳ ai cũng có quyền đọc và ghi dữ liệu trên Blockchain

**Private:** Người dùng chỉ được quyền đọc dữ liệu, không có quyền ghi vì điều này thuộc về bên tổ chức thứ ba tuyệt đối tin cậy.

**Permissioned (hay còn gọi là Consortium):** một dạng của Private nhưng bổ sung thêm 1 số tính năng khác, đây là sự kết hợp giữa Public và Private.

* Blockchain có các đặc điểm nổi bật sau:
* **Không thể làm giả, không thể phá hủy các chuỗi Blockchain**
* **Bất biến**: dữ liệu trong Blockchain không thể sửa và sẽ lưu trữ mãi mãi.
* **Bảo mật:** Các thông tin, dữ liệu trong Blockchain được phân tán và an toàn tuyệt đối
* **Minh bạch**: Ai cũng có thể theo dõi dữ liệu Blockchain đi từ địa chỉ này tới địa chỉ khác và có thể thống kê toàn bộ lịch sử trên địa chỉ đó.
* Các giao dịch chỉ được thực hiện, xác định khi có sự đồng thuận của tất cả người dùng trên hệ thống và sẽ lưu lại mãi mãi, không thể bị làm giả hoặc gỡ bỏ bằng bất kỳ thủ thuật nào. Điều này làm cho việc đánh cắp thông tin là bất khả thi
* Hợp đồng thông minh**:** là hợp đồng kỹ thuật số được nhúng vào đoạn code if-this-then-that (IFTTT), cho phép chúng tự thực thi mà không cần bên thứ ba
* **Cơ chế hoạt động**
* Ứng dụng được biết đến và thảo luận nhiều nhất về công nghệ Blockchain chính là**đồng tiền điện tử** Bitcoin.

Để theo dõi số lượng Bitcoin mà mỗi người sở hữu trong các tài khoản nhất định và theo dõi các giao dịch phát sinh từ đó thì chúng ta cần đến một **cuốn sổ kế toán, trong trường hợp này chính là Blockchain.**

Tệp sổ cái này không được lưu trữ trong một máy chủ trung tâm, như trong một ngân hàng hoặc trong một trung tâm dữ liệu mà ngược lại nó được phân phối trên toàn thế giới thông qua một mạng lưới các máy tính ngang hàng với vai trò lưu trữ dữ liệu và thực thi các tính toán.

 Mỗi máy tính này đại diện cho một “nút” của mạng lưới Blockchain và mỗi nút đều có một bản sao của tệp sổ cái này.

Công nghệ chuỗi khối Blockchain là một cuốn sổ cái ghi chép và lưu trữ thông tin các giao dịch. Blockchain, có thể nói đây là một cơ sở dữ liệu được tổ chức thành liên kết dạng chuỗi của các khối thông tin (block), cho phép phát triển và mở rộng theo thời gian, nghĩa là bất cứ khi nào có những dữ liệu mới thì sẽ hình thành thêm các khối mới

**Cấu trúc dữ liệu cần lưu trữ trong các block:**

Một khối Blockchain gồm 4 thành phần:

* Mã băm: là mã đại diện cho mỗi một block, và không trùng với bất kì khối nào khác trong chuỗi
* Dữ liệu giao dịch: một đoạn văn bản hay dữ liệu bất kì, là dữ liệu giao dịch quan trọng
* Dấu thời gian: là mốc thời gian mà block được tạo ra
* Mã băm của khối trước

Mã băm là gì:

Là một thuận toán xáo trộn mã nhị phân của dữ liệu sao cho mã đầu vào sau khi băm dù dài hay ngắn cũng trở thành một chuỗi cố định và không trùng lặp

Ngoài ra, hash mật mã còn mang một số đặc điểm khác như tính tránh va chạm (xảy ra khi hai giá trị khác nhau nhưng khi chạy qua hàm băm lại trả về hai kết quả giống nhau), tính hiệu quả (thời gian tính toán những giá trị băm phải nhanh) và cả tính nhạy cảm (chỉ cần sự thay đổi nhỏ trong giá trị ban đầu có thể thay đổi hoàn toàn giá trị băm).

Có 2 thuật toán tạo mã băm: MD5, SHA256.

MD5: không an toàn cho các giao dịch quan trọng, vì nó dễ xảy ra xung đột khi hoạt động

SHA256:

* **Cơ chế liên kết**

Blockchain lưu trữ thông tin trong các khối, các khối liên kết với nhau theo thời gian thực để tạo thành 1 chuỗi.

Thông tin đã được lưu trữ trong khối, thì khối đó ko cho phép bất cứ ai chỉnh sửa nữa. vì vậy để chỉnh sửa thông tin đó thì chỉ có duy nhất một cách là ghi thông tin lên 1 khối mới và kèm theo thông tin của người chỉnh sửa và thời gian cự thể.

* **Các nguyên tắc của blockchain:**
* Không xóa hoặc chỉnh sửa Block cũ.
* Chỉ thêm Block mới.
* Hạn chế:
* **Thông tin lưu trữ mãi mãi và không thể sửa được**
* Vấn đề của blockchain là nó lưu trữ mọi thứ mãi mãi, và nó có thể gây ra một sự lãng phí lớn về không gian lưu trữ. Càng sử dụng lâu, càng nhiều “rác”.
* Có 2 cách để ứng dụng Blockchain cho sản phẩm của mình. 1 là ứng dụng 1 mạng lưới Blockchain đã có sẵn – lựa chọn thường thấy là mạng ETH. Cách thứ 2 là xây dựng mạng Blockchain của riêng mình.
* Nếu mạng tự xây dựng không đủ độ lớn cần thiết, nguy cơ bị tấn công quá bán là rất cao
* ứng dụng của mã băm
* Tính bảo mật cao
* Các hàm băm mật mã có nhiều ứng dụng trong an toàn thông tin. Nó được sử dụng nhiều trong chữ ký số, mã xác thực thông điệp (MAC) và các hình thức xác thực khác. Ngoài ra, chúng cũng có thể được sử dụng như các hàm băm thông thường, để lập chỉ mục dữ liệu trong bảng băm, lấy đặc trưng của dữ liệu, phát hiện dữ liệu trùng lặp hoặc làm tổng kiểm tra để phát hiện lỗi các dữ liệu ngẫu nhiên
* **Nâng cao hiệu quả chữ ký số:**hầu như tất cả các lược đồ chữ ký số đều yêu cầu tính toán bản tóm lược của thông điệp bằng các hàm băm mật mã. Điều này cho phép việc tính toán và tạo chữ ký được thực hiện trên một khối dữ liệu có kích thước tương đối nhỏ và cố định thay vì trên toàn bộ văn bản dài.

## Mã hóa

Là một phương pháp bảo vệ dữ liệu bằng cách chuyển đổi dữ liệu từ dạng này sang dạng khác mà chỉ có người có quyền truy cập vào khóa giải mã hoặc có mật khẩu mới có thể đọc được nó. Dữ liệu được mã hóa thường gọi là ciphertext, dữ liệu thông thường, chưa được mã hóa thì gọi là plaintext.

### Chức năng chính của mã hóa dữ liệu

Mục đích của việc mã hóa dữ liệu là để bảo vệ dữ liệu kỹ thuật số vì nó được lưu trữ trên các hệ thống máy tính và được truyền qua internet hoặc các mạng máy tính khác. Tiêu chuẩn mã hóa dữ liệu lỗi thời (DES) đã được thay thế bằng các thuật toán mã hóa hiện đại đóng vai trò quan trọng trong việc bảo mật hệ thống CNTT và truyền thông.

Các thuật toán này cung cấp tính bảo mật và thúc đẩy các sáng kiến bảo mật quan trọng bao gồm xác thực, tính toàn vẹn và không thu hồi. Xác thực cho phép xác minh nguồn gốc của dữ liệu, tính toàn vẹn cung cấp bằng chứng cho thấy nội dung dữ liệu không thay đổi kể từ khi nó được gửi đi và không thu hồi đảm bảo rằng người gửi không thể hủy việc gửi dữ liệu.

### Các phương pháp mã hóa dữ liệu phổ biến

* Mã hóa đối xứng

Phương pháp mã hóa dùng cùng một key cho việc mã hóa và giải mã. Trong khi mã hóa đối xứng nhanh hơn nhiều so với mã hóa không đối xứng, người gửi cần phải trao đổi khóa giải mã với người nhận để giải mã dữ liệu. Thuật toán thường dùng: DES, AES.

* Mã hóa không đối xứng

Sử dụng hai khóa khác nhau, public key được dùng để mã hóa có thể được chia sẻ công khai trong khi private key được bên nhận giữ lại. Trước khi gửi tin nhắn, người gửi sẽ mã hóa dữ liệu bằng mã hóa bất đối xứng với những key nhận được từ người nhận. Người nhận sẽ giải mã dữ liệu nhận được bằng thuật toán được sử dụng ở bên người gửi, với key giải mã là private key. Thuật toán thường dùng: RSA.

* Mã hóa một chiều (hash)

Được sử dụng khi không cần giải mã dữ liệu đã được mã hóa. Thuật toán thường dùng: MD5, SHA.

## Hash (hàm băm) là gì và cách thức hoạt động như thế nào?

Hash (hàm băm) là một trong những phần quan trọng nhất của thứ công nghệ giúp giữ an toàn cho mạng lưới Bitcoin

### Hash là gì?

Về cơ bản hashing là quá trình biến một dữ liệu đầu vào có độ dài bất kỳ thành một chuỗi đầu ra đặc trưng có độ dài cố định. Hashing được thực hiện thông qua hàm băm (hash function).

Một cách tổng quát hàm băm là bất kỳ hàm nào có thể được sử dụng để ánh xạ dữ liệu có kích thước tùy ý thành các giá trị kích thước cố định. Các giá trị được trả về bởi hàm băm được gọi là giá trị băm, mã băm, thông điệp băm, hoặc đơn giản là “hash”.

Ví dụ, khi bạn download một video trên YouTube có dung lượng 50 MB và thực hiện hashing trên nó bằng thuật toán băm SHA-256, thì đầu ra bạn thu được sẽ là một giá trị băm có độ dài 256 bit. Tương tự, nếu bạn lấy một tin nhắn văn bản có dung lượng 5 KB, để hashing bằng SHA-256 thì giá trị băm đầu ra bạn thu được vẫn sẽ là 256 bit.

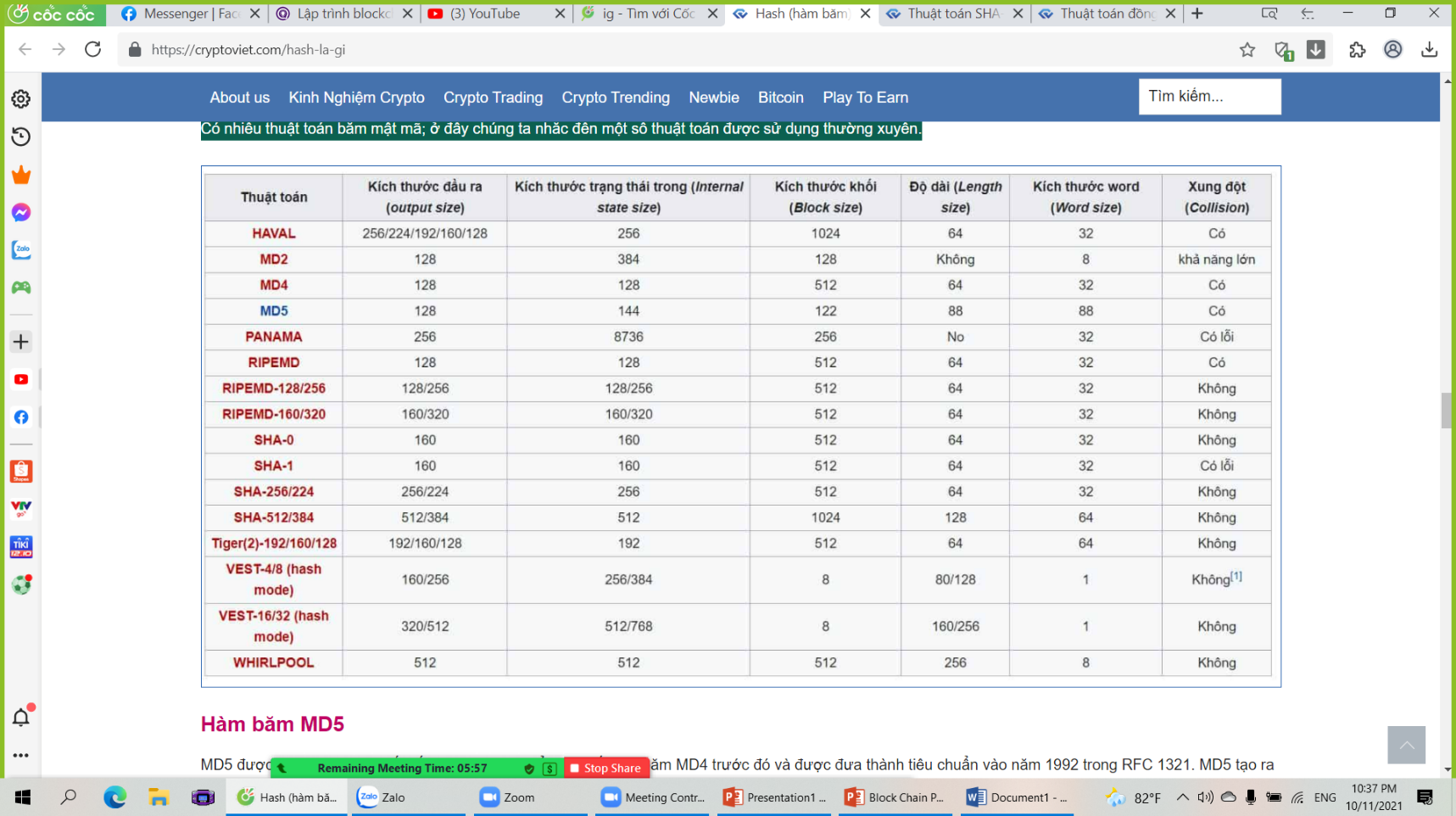
Như bạn có thể thấy, trong trường hợp SHA-256, cho dù đầu vào của bạn lớn hay nhỏ như thế nào đi chăng nữa, thì đầu ra bạn nhận được sẽ luôn có độ dài 256 bit cố định. Điều này trở nên quan trọng khi bạn xử lý một lượng lớn dữ liệu và giao dịch. Khi đó, thay vì bạn phải xử lý toàn bộ lượng dữ liệu đầu vào (có thể có kích thước rất lớn), bạn chỉ cần xử lý và theo dõi một lượng dữ liệu rất nhỏ là các giá trị băm.

Trong blockchain, các giao dịch có độ dài khác nhau sẽ được băm thông qua một thuật toán băm nhất định và tất cả đều cho đầu ra có độ dài cố định bất kể độ dài của giao dịch đầu vào là bao nhiêu. Chẳng hạn, Bitcoin sử dụng thuật toán SHA-256 để băm các giao dịch cho kết quả đầu ra có độ dài cố định là 256 bit (32 byte) cho dù giao dịch chỉ là một từ hoặc giao dịch phức tạp với lượng dữ liệu khổng lồ. Điều làm cho việc theo dõi các giao dịch trở nên dễ dàng hơn khi truy xuất và theo dõi lại các giá trị băm. Kích thước của hàm băm sẽ phụ thuộc vào hàm băm được sử dụng.

Kỹ thuật hashing thường được sử dụng và có ứng dụng rộng rãi nhất trong việc đảm bảo tính toàn vẹn cho dữ liệu trong blockchain là các hàm băm mật mã (cryptographic hash function) chẳng hạn như SHA-1. SHA-2. SHA-3, SHA-256…Sỡ dĩ như vậy là do các hàm băm mật mã có một số tính chất quan trọng phù hợp cho việc đảm bảo an toàn dữ liệu.

1. Những dạng Hash cơ bản

Có nhiều thuật toán băm mật mã; ở đây chúng ta nhắc đến một số thuật toán được sử dụng thường xuyên.



1. SHA-2

SHA-2 là một tập hợp các hàm băm mật mã được thiết kế bởi Cơ quan an ninh quốc gia Hoa Kỳ (NSA), được xuất bản lần đầu tiên vào năm 2001. Chúng được xây dựng bằng cấu trúc Merkle–Damgård, chức năng nén một chiều của nó được xây dựng bằng cấu trúc Davies–Meyer từ một hệ mật mã khối chuyên dụng.

SHA-2 về thực chất bao gồm hai thuật toán băm: SHA-256 và SHA-512. SHA-224 là một biến thể của SHA-256 với các giá trị khởi tạo và đầu ra bị cắt bỏ khác nhau. SHA-384 và SHA-512/224 và SHA-512/256 ít được biết đến là tất cả các biến thể của SHA-512. SHA-512 an toàn hơn SHA-256 và thường nhanh hơn SHA-256 trên các máy 64 bit như AMD64.

Do có nhiều phiên bản thuật toán khác nhau do đó kích thước đầu ra của họ SHA-2 cũng khác nhau tùy theo thuật toán. Phần mở rộng của tên phía sau tiền tố “SHA” chính là độ dài của thông điệp băm đầu ra. Ví dụ với SHA-224 thì kích thước đầu ra là 224 bit (28 byte), SHA-256 tạo ra 32 byte, SHA-384 tạo ra 48 byte và cuối cùng là SHA- 512 tạo ra 64 byte. Và chúng ta có thể đã biết rằng Bitcoin sử dụng hàm băm SHA-256 là một phiên bản trong họ SHA-2 này.

1. SHA-3
2. BLAKE2
3. Hàm băm MD5

### Ứng dụng của Hash

Hashing nói chung và hàm băm mật mã nói riêng có rất nhiều ứng dụng khác nhau trong thực tế. Dưới đây là một số ứng dụng phổ biến nhất của nó:

1. Hashing trong định danh tệp hoặc dữ liệu

Giá trị băm cũng có thể được sử dụng như một phương tiện để định danh tập tin một cách đáng tin cậy. Một số hệ thống quản lý mã nguồn, như Git, Mercurial hay Monotone, sử dụng giá trị sha1sum của nội dung tệp, cây thư mục, thông tin thư mục gốc, v.v. để định danh chúng.

Giá trị băm cũng được sử dụng để xác định các tệp trên các mạng chia sẻ tệp ngang hàng nhằm cung cấp đầy đủ thông tin để định vị nguồn gốc của tệp, xác minh nội dung tệp tải xuống. Giá trị ứng dụng của chúng còn được mở rộng ra khi áp dụng các cấu trúc dữ liệu bổ sung như danh sách băm hoặc cây băm (Merkle Tree).

Tuy nhiên, so với các hàm băm tiêu chuẩn, các hàm băm mật mã có xu hướng phức tạp và đòi hỏi nhiều tài nguyên tính toán hơn nhiều. Do đó, chúng có xu hướng được sử dụng trong các trường hợp khi người dùng cần tự bảo vệ tính an toàn cho thông điệp trước các khả năng bị sửa đổi, hay giả mạo như các ứng dụng ở dưới đây:

1. Hashing trong xác minh tính toàn vẹn của thông điệp hoặc tập tin

Một ứng dụng quan trọng nhất của hashing là xác minh tính toàn vẹn của thông điệp.Chúng ta khá quen thuộc với các ứng dụng này. Khi download một phần mềm hoặc tệp tin nào đó trên một số trang web, ta được cung cấp kèm theo các mã băm MD5 hoặc SHA1. Khi đó sau khi tải về tập tin, chúng ta có thể tính và so sánh giá trị băm của tệp tải về với giá trị băm được cung cấp trên web, nếu có sự sai khác tức là tệp tin chúng ta tải về đã bị sửa đổi.

1. Hashing trong tạo và xác nhận chữ ký

Hầu như tất cả các lược đồ chữ ký số đều yêu cầu tính toán bản tóm lược của thông điệp bằng các hàm băm mật mã. Điều này cho phép việc tính toán và tạo chữ ký được thực hiện trên một khối dữ liệu có kích thước tương đối nhỏ và cố định thay vì trên toàn bộ văn bản dài. Tính chất toàn vẹn thông điệp của hàm băm mật mã được sử dụng để tạo các lược đồ chữ ký số an toàn và hiệu quả.

1. Hashing và Bằng chứng công việc (Proof of Work)

Bằng chứng công việc (Proof of Work) là một biện pháp kinh tế để ngăn chặn các cuộc tấn công từ chối dịch vụ và các hành vi lạm dụng dịch vụ khác như spam bằng cách yêu cầu người dùng dịch vụ thực hiện một số công việc nhất định, thường đòi hỏi nhiều thời gian xử lý. Bằng chứng công việc cần đảm bảo tính bất đối xứng tức là: công việc phải có độ khó vừa phải (nhưng khả thi) về phía người dùng nhưng dễ kiểm chứng đối với nhà cung cấp dịch vụ.

Hệ thống Proof of Work đầu tiên được đề xuất là Hashcash. Hashcash sử dụng quá trình hashing như một phần để chứng minh rằng công việc đã được thực hiện nhằm cho phép gửi email đi, tránh các email spam. Công việc trung bình mà người dùng cần thực hiện để tìm thư hợp lệ cấp số nhân với số bit 0 cần thiết trong giá trị băm, trong khi người nhận có thể xác minh tính hợp lệ của email chỉ bằng cách thực hiện một phép tính băm duy nhất. Trong Hashcash, người gửi được yêu cầu tạo một tiêu đề có giá trị băm SHA-1 160 bit trong đó có 20 bit đầu tiên là bit 0. Khi đó, người gửi trung bình sẽ phải thử khoảng lần để tìm ra tiêu đề hợp lệ trước khi gửi đi.

Hệ thống này được thừa kế trong Bitcoin, nền tảng blockchain đầu tiên. Việc tính các giá trị băm giúp mở khóa phần thưởng khai thác trong Bitcoin. Các thành viên trong mạng được yêu cầu tìm một giá trị sao cho sự kết hợp của nó với thông điệp gốc (tập hợp các giao dịch) có giá trị băm bắt đầu bằng một số bit không (xác định bởi độ khó của việc khai thác và được điều chỉnh thường xuyên bằng phần mềm).

1. Hashing trong blockchain

Một số hàm băm mật mã đang được sử dụng rộng rãi như đã kể ở trên:

* SHA 256 hiện đang được sử dụng bởi Bitcoin.
* Keccak-256 hiện đang được Ethereum sử dụng.

Các hàm băm này không những được sử dụng để tạo ra bằng chứng công việc (Proof of Work) mà còn để làm định danh cho các khối, hoặc kết hợp với mật mã khóa công khai để tạo ra định danh cho người dùng trên mạng.

## Thuật toán SHA-256 là gì?

SHA-256 là một trong những hàm băm mạnh nhất hiện nay và được áp dụng vào hệ thống tiền ảo Bitcoin

### Mã hoá SHA là gì?

SHA (Secure Hash Algorithm) bao gồm 5 thuật toán được chấp nhận bởi FIPS – Tiêu chuẩn Xử lý Thông tin Liên bang, dùng để chuyển một đoạn dữ liệu nhất định thành một đoạn dữ liệu có chiều dài không đổi với xác suất khác biệt cao. 5 thuật toán đó bao gồm:

* SHA-1 (trả lại kết quả dài 160 bit)
* SHA-224 (trả lại kết quả dài 224 bit)
* SHA-256 (trả lại kết quả dài 256 bit)
* SHA-384 (trả lại kết quả dài 384 bit)
* SHA-512 (trả lại kết quả dài 512 bit)

Những thuật giải này được gọi là “an toàn” bởi vì, theo nguyên văn của chuẩn mực FIPS 180-2 phát hành ngày 1/8/2002:

Đối với một giá trị nhất định được tạo nên bởi một trong những thuật toán SHA, việc tính toán là không khả thi để:

1. Tìm một thông điệp tương ứng với thông điệp đã được mã hóa
2. Tìm được hai đoạn dữ liệu khác nhau có cùng kết quả băm.

Bất cứ thay đổi nào trên đoạn dữ liệu gốc, dù nhỏ, cũng sẽ tạo nên một giá trị băm hoàn toàn khác với xác suất rất cao.

### SHA-256 là gì?

SHA-256 là Thuật toán băm bảo mật 256 bit và dùng để tạo ra các hàm băm không thể đảo ngược và duy nhất. Số lượng hàm băm có thể có càng lớn, thì xác suất để hai giá trị sẽ tạo ra cùng một giá trị băm càng nhỏ.

Ví dụ:

SHA-256 là một nhánh của hàm băm mật mã SHA-2 được sử dụng trong nhiều phần khác nhau của mạng Bitcoin:

|  |  |
| --- | --- |
| **DỮ LIỆU** | **CHUỖI MÃ HÓA** |
| Bitcoin Vietnam News | f3ad777234a24bfacbd8123d6ea0a1961e8539b74b5fa8f2f371ea2cf7b21215 |
| cryptoviet.com | cbab80bf094869581bb45557b64a8db0b8bab8c8817b5facd0c975d9a5a47805 |

* Khai thác sử dụng SHA-256 là thuật toán Proof of Work
* SHA-256 được sử dụng trong việc tạo ra các địa chỉ bitcoin để cải thiện an ninh và bảo mật.

Thuật toán SHA-256 tạo ra một mã băm có cố định 256-bit (32-byte) gần như duy nhất. Một chuỗi băm được tạo ra không thể được tính toán trở lại. Điều này làm cho nó phù hợp để xác nhận mật khẩu, thách thức xác thực hàm băm, chống giả mạo, chữ ký số.

SHA-256 là một trong những hàm băm kế tiếp đối sau SHA-1 và là một trong những hàm băm mạnh nhất hiện tại.

### Ứng dụng của SHA-256

Ứng dụng nổi tiếng nhất của SHA phải kể đến là hệ thống Tiền tệ Bitcoin, một hệ thống giao dịch tiền tệ phân tán trên mạng internet sử dụng mã SHA-256 để xác thực các giao dịch và lưu trữ dạng chuỗi các sự kiện lịch sử theo thời gian được liên kết với nhau bởi các mã xác thực BlockChain.

### Liệu có thể tấn công Brute Force vào SHA-256 không?

Cách dùng phổ biến của mã một chiều SHA là tạo ra chữ ký của thông điệp bằng cách tính hàm băm của chuỗi ghép từ thông điệp cần xác thực với một khóa bí mật, khóa bí mật nói chung là rất dài:

Chữ ký = SHA256(“thông điệp cần xác thực”, “khóa bí mật”) = hash

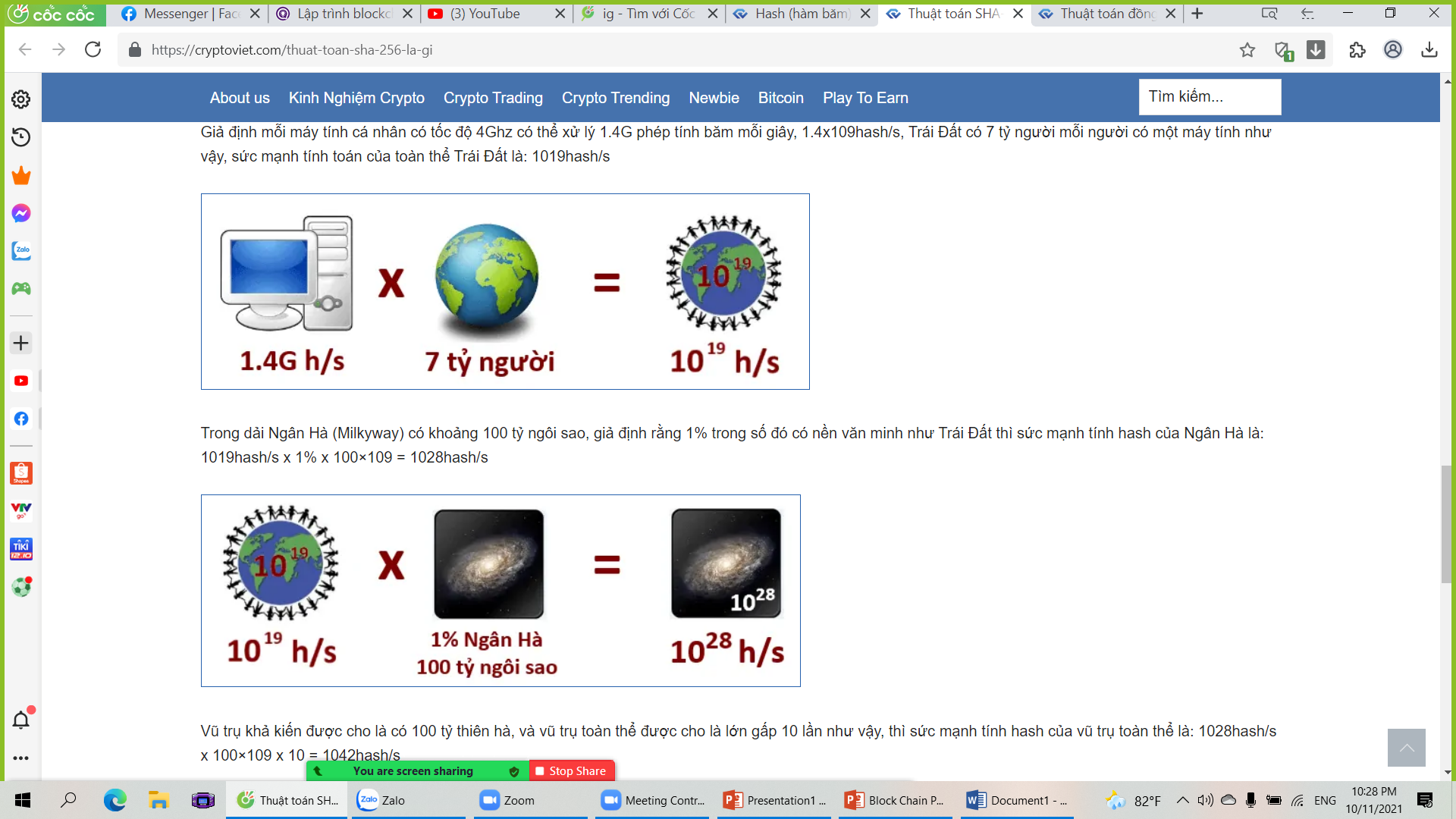
Với tính chất của mã một chiều, chữ ký có thể được công khai, thông điệp có thể công khai, nhưng thành phần khóa bí mật không thể truy ngược lại được vì suy ngược lại toàn văn dữ liệu gốc từ chữ ký là không thể.

Cách duy nhất để truy ngược lại toàn văn dữ liệu gốc là thử mọi khả năng có thể của khóa bí mật và cách khóa được ghép với thông điệp cần xác thực, điều này cũng đồng nghĩa với với việc kiểm tra mọi khả năng của đầu vào để tìm ra chữ ký giống với chữ ký đã biết, phương pháp này gọi là Brute Force

Vì mỗi thay đổi rất nhỏ ở dữ liệu gốc, dù nhỏ, cũng sẽ tạo nên một giá trị băm hoàn toàn khác với xác suất rất cao không thể dự báo trước, nên việc kiểm tra này chính là kiểm tra trên toàn bộ không gian của giá trị băm, tức là BruteForce với 2256 khả năng, tương ứng với:

2256=115792089237316195423570985008687907853269984665640564039457584007913129639936 > 1078

Giả định mỗi máy tính cá nhân có tốc độ 4Ghz có thể xử lý 1.4G phép tính băm mỗi giây = 1.4x109hash/s, Trái Đất có 7 tỷ người mỗi người có một máy tính như vậy, sức mạnh tính toán của toàn thể Trái Đất là: 1019hash/s



## PROOF OF WORK

### POW là gì?

Là một thuật toán đồng thuận được giới thiệu bởi Bitcoinvà được sử dụng rộng rãi bởi nhiều đồng tiền điện tử khác. Quá trình này được biết đến với cách gọi thông thường là “đào” và các nút trên mạng được gọi là “thợ đào”.

Proof of Work yêu cầu thợ đào phải giải các bài toán mật mã phức tạp để hợp thức hoá các khối (block) trong blockchain và nhận lại phần thưởng dưới dạng coin hay token mới phát hành.

Đúng với cái tên gọi “bằng chứng công việc”, bạn sẽ phải “làm việc” mới được thưởng/trả công, bằng cách:

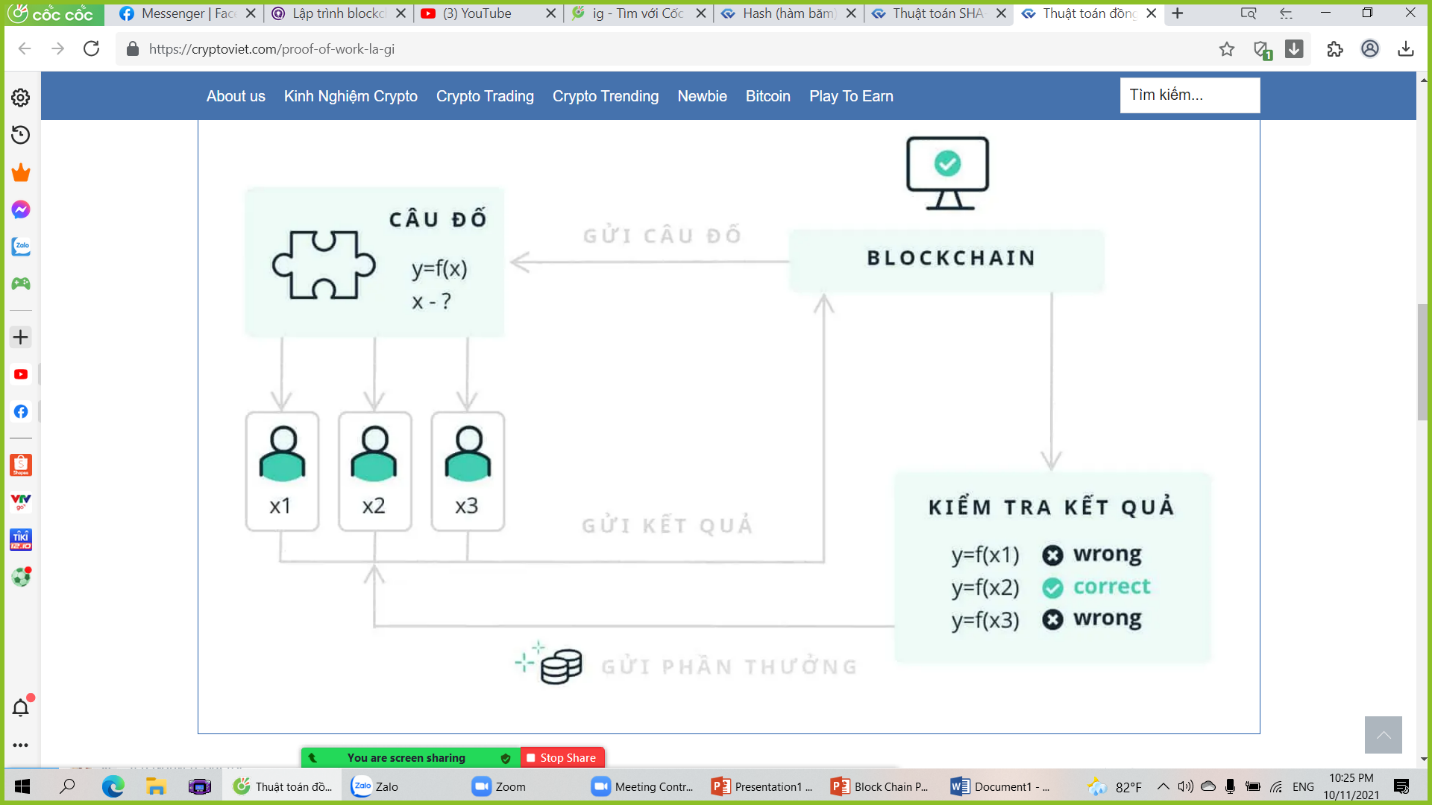
* Cung cấp máy móc, sức mạnh của máy.
* Tiêu thụ điện để giải quyết các thuật toán vô cùng phức tạp. Ai có càng nhiều máy, máy càng mạnh, đốt nhiều điện sẽ giải các thuật toán nhanh hơn và chính xác hơn.
* Sau đó, hệ thống sẽ chọn ra đáp án tốt nhất. Người nào đưa ra đáp án này sẽ trở thành Validator (người xác nhận). Và người đó có quyền khai thác block mới, xác nhận các giao dịch trong block đó.
* Cuối cùng là nhận phần thưởng chính là coin/token.

### Nguyên lí hđ

Khi thực hiện giao dịch trên Blockchain, nó sẽ được gom vào một Block cùng một số giao dịch khác. Các thợ đào sẽ sử dụng hệ thống máy đào gồm nhiều máy tính mạnh để xác minh giao dịch.

Một câu đố toán học phức tạp sẽ được hệ thống đưa ra. Nhiệm vụ của thợ đào là sử dụng sức mạnh của hệ thống đào tìm ra câu trả lời, sau khi tìm được sẽ thông báo cho các thợ đào còn lại. Khi phần lớn thành viên xác nhận đó là câu trả lời đúng, Block mới sẽ được tạo ra, giao dịch được xác nhận.

Khi hoàn thành, thợ đào sẽ nhận được phần thưởng là phí giao dịch và phần thưởng khối. Tuy nhiên, đây là quá trình sử dụng rất nhiều tài nguyên, điện, thời gian.



### Đánh giá ưu nhược điểm của Proof of Work

1. Ưu điểm

Thuật toán Proof of Work có khả năng ngăn chặn các cuộc tấn công DDoS và các tác động khác của các phần mềm tiền điện tử của thợ mỏ. Do các chức năng hạn chế quyền hạn, thuật toán Proof of Work áp đặt nhiều chính sách nhất định đối với những người tham gia. Những người tham gia cho dù nắm giữ một số tiền lớn cũng không thể có quyền quyết định cho cả mạng lưới. Bạn cần có khả năng tính toán để tìm ra các khối mới.

1. Nhược điểm

Chi phí dành cho thực hiện thuật toán Proof of Work là rất lớn. Bởi lẽ với các tính toán phức tạp như vậy, thiết bị máy tính thông thường không thể đảm nhận được, cần bỏ nhiều chi phí cho việc đầu tư thiết bị chuyên dụng, chưa kể đến các loại chi phí quản lý và khai thác các mỏ. Các máy tính này cũng tiêu thụ cực nhiều năng lượng, chi phí vì thế được đội lên nhiều hơn. Điều này gây ra sự gia tăng tập trung của hệ thống.

Trong khi thực hiện các công việc tạo khối, các thuật toán không cần thiết và vô dụng là điều hoàn toàn bình thường, các kết quả có thể chẳng bao giờ được sử dụng. Các thợ mỏ sẽ muốn bảo đảm an toàn. thế nhưng tỷ lệ tấn công có thể lên đến 51%.

### Tầm quan trọng của Proof of Work

Proof of Work mang tới sự tin tưởng trong môi trường phi tập trung. Khi các thợ đào đồng ý cạnh tranh để tạo Block mới và nhận thưởng, họ đã ngầm tuân theo quy tắc của hệ thống, thay vì cố gắng thao túng.

Với thuật toán điều chỉnh độ khó ở mỗi Block, đảm bảo các Block mới sẽ được tạo ra với tốc độ ổn định. Giúp duy trì nguồn cung và khuyến khích thợ đào hoạt động.

Tuy nhiên, chính việc sử dụng các nguồn lực hữu hạn như thời gian, năng lượng, sức mạnh tính toán đã làm hạn chế khả năng mở rộng của PoW.

### Kết luận

Mặc dù có một số thành tích, PoW được coi là một giao thức đồng thuận thiếu sót, đặc biệt khi xem xét lượng năng lượng được tiêu thụ khi chạy giao thức. Ví dụ, nó đã được báo cáo rằng một giao dịch Bitcoin duy nhất, bằng cách sử dụng PoW, có thể tiêu thụ nhiều điện năng như một hộ gia đình trung bình của Hà Lan trong hai tuần.

Kết quả là, đã có một sự dịch chuyển đến các giao thức đồng thuận tiến bộ hơn như giao thức đồng thuận Delegated Proof of Stake

CHUYÊN MỤC TRẢ LỜI CÂU HỎI :

**cho mình hỏi sự khác nhau giữa interpreter và compiler (thông dịch vs biên dịch)**

1. Interpreter chuyển đổi một đoạn mã lệnh, thực thi nó sau đó mới lặp lại quá trình với đoạn mã lệnh tiếp theo. Trong khi đó, Compiler chuyển đổi toàn bộ mã lệnh một lần, sau đó mới thực thi. 2. Compiler khởi tạo báo cáo lỗi sau khi chuyển đổi toàn bộ mã lệnh trong khi Interpreter sẽ dừng thông dịch nếu phát hiện ra lỗi. 3. Trong cùng một tiến trình, Compiler sẽ mất một lượng thời gian khi phân tích và xử lý ngôn ngữ bậc cao lớn hơn so với Interpreter. 4. Thời gian phân tích và xử lý một đoạn mã lệnh được thực thi tổng thể của compiler thì nhanh hơn so với interpreter

**Câu hỏi cho nhóm 5: Nếu một khối bị thay đổi thì các khối sau có bị ảnh hưởng gì không?**

Khi thay đổi nội dung 1 khối thì mã hash của khối đó sẽ thay đổi và do các khối được liên kết với nhau theo mã băm nêu chuỗi khối sẽ bị đứt.

**Câu hỏi cho nhóm 5:Các bạn có thể cho ví dụ cụ thể về blockchain private ? blockchain private này có cho phép các nút gian dối không ?**

* Blockchain riêng tư có tổ chức điều hành và quản lý.
* Chỉ có cá nhân sử dụng Blockchain đó mới có thể truy cập và sửa đổi.
* Mất private key tài khoản sẽ thất lạc vĩnh viễn.

**Nhóm 5: Các bạn có thể so sánh các thuộc tính: tính bất biến, hiệu suất, tập trung, tiến trình đồng thuận của các loại blockchain public, private, permissioned được không**

**cho mình hỏi nhóm 5: Có chắc chắn là các block ko thể thay đổi ko? nếu có thể thay đổi thì điều kiện cần là gì?**

Nếu có thể thì bạn sẽ cần tính toán lại mã băm cho các khối phía sau, đồng thời tính toán lại trên ít nhất 51% máy tính còn lại tham gia mạng.

**Câu hỏi cho nhóm 5: Cho mình hỏi nếu như 1 hacker nằm trong khối blockchain đó khi có giao dịch xảy ra chẳng hạng như giao dịch tiền tệ hacker muốn lấy cái tiền từ người gửi thì hacker có thực hiên hay không ?**

khi bạn muốn thay đổi dữ liệu giao dịch của Bitcoin để đánh cắp tiền, bạn sẽ phải có lượng máy tính đủ lớn để áp đảo mạng. Nhưng khi làm như vậy, bạn đã vô tình cô lập mình ra khỏi cộng đồng vì đồng tiền chỉ có giá trị khi có nhiều người dùng. Nếu bạn hack theo cách đó thì đồng tiền sẽ sụt giá trị và không mang lại ý nghĩa gì cả. Thay vào đó, nếu đầu tư các máy tính tham gia vào mạng này, bạn sẽ được thưởng Bitcoin cho mỗi block được ghi nhận vào khối. Điều này có lợi hơn là tấn công chiếm đoạn Bitcoin.

**Nhóm 5 cho mình hỏi: Blockchain có ưu điểm gì so với các hình thức lưu trữ khác**

Đặc điểm của Blockchain:

-Không thể làm giả, không thể phá hủy

-Bất biến

-Bảo mật

-Minh bạch

-Hợp đồng thông minh

**Nhóm 5 cho mình hỏi: Các bạn có nói đến "Pow" vậy các bạn biết đến "Pos" không? Sự khác biệt là gì?**

Đều là thuật toán đồng thuận

Khác biệt: Trong khi ở Proof of Stake, các Node phải đặt cọc coin để tham gia xác nhận các giao dịch trên block nhằm xác minh danh tính và bảo đảm block tiếp theo sẽ hợp lệ. Nếu đúng thì họ sẽ nhận được phần thưởng còn nếu sai thì người đặt cọc sẽ mất số tiền mình cọc ra trước đó. Thì Proof of Work phải giải mã các bài toán mật mã sử dụng sức mạnh tính toán để xác minh các giao dịch.

**nhóm 5:các bạn có thể cho mình biết vi dụ về hợp đồn thông minh là như thế nào không ?**

Chiếc máy bán hàng tự động là một hình ảnh ví dụ trực quan đơn giản nhất về Hợp đồng thông minh. Khi người mua hàng nhập lệnh lựa chọn sản phẩm và đút tiền vào khe nhận tiền, sản phẩm mà người khách mong muốn sẽ được tự động chuyển ra [ii].

**Nhóm 5: Cho mình hỏi dùng trên mã nguồn mở và free làm sao nhóm có thể khẳng định 100% thông tin là hoàn toàn riêng tư được, và nhóm dựa vào yếu tố nào để chỉ ra rằng bên cung cấp dịch vụ sẽ không bán thông tin khi đủ lớn?**