



Giới thiệu thuật toán

SHA-256

- SHA-256 (Secure Hash Algorithm 256-bit) là một thuật toán băm mật mã thuộc nhóm SHA-2 (Secure Hash Algorithm 2), được phát triển bởi Cơ quan An ninh Quốc gia Mỹ (NSA) và công bố vào năm 2001. SHA-256 được sử dụng rộng rãi trong các ứng dụng bảo mật, đặc biệt là trong các hệ thống blockchain như Bitcoin.

Nguyễn Hoàng Hải Văn - 23521771
Lê Anh Tuấn - 23521712

Các đặc điểm chính của SHA-256

1

Độ dài băm cố định

SHA-256 tạo ra một giá trị băm dài 256 bit (hay 64 ký tự hex), bất kể độ dài của đầu vào (message) là bao nhiêu.

Ví dụ: Một đầu vào bất kỳ (dù dài hay ngắn) sau khi băm bằng SHA-256 sẽ luôn cho ra một chuỗi 64 ký tự.

3

Tính không thể va chạm

Rất khó (gần như không thể) để tìm hai đầu vào khác nhau tạo ra cùng một giá trị băm. Đây là yếu tố rất quan trọng để đảm bảo tính toàn vẹn của dữ liệu.

2

Một chiều

Được gọi là hàm băm một chiều vì rất dễ tính toán giá trị băm từ đầu vào nhưng không thể đảo ngược để tìm lại dữ liệu ban đầu từ giá trị băm. Đây là đặc điểm quan trọng trong bảo mật.

4

Hàm băm an toàn

SHA-256 bảo vệ dữ liệu khỏi việc bị giả mạo hoặc thay đổi, vì ngay cả một thay đổi nhỏ trong đầu vào cũng sẽ dẫn đến sự thay đổi hoàn toàn trong kết quả băm.



Made with Gamma

Cách thức hoạt động của SHA-256

SHA-256 hoạt động thông qua quá trình gọi là "hashing", tức là xử lý một thông điệp đầu vào và tạo ra một "digest" (dữ liệu băm) có độ dài cố định. Quá trình này bao gồm nhiều bước phức tạp, bao gồm việc chia thông điệp thành các khối nhỏ, áp dụng các phép toán số học, logic và kết hợp chúng qua nhiều vòng lặp để tạo ra giá trị băm cuối cùng.

Các bước cơ bản của SHA-256

Chuẩn hóa thông điệp

Đầu vào sẽ được chia thành các khối 512-bit, sau đó được bổ sung dữ liệu để đảm bảo độ dài của nó là bội số của 512. Bổ sung bao gồm một bit 1 và sau đó là các bit 0 cho đến khi thông điệp đạt kích thước phù hợp.

Chia thành các khối nhỏ

Sau khi chuẩn hóa, thông điệp sẽ được chia thành các khối 512-bit. Mỗi khối sẽ được xử lý riêng biệt.

Khởi tạo giá trị hằng số

SHA-256 sử dụng 8 hằng số (được gọi là "initial hash values"), là các số thập phân đã được xác định trước.

Vòng lặp tính toán

Mỗi khối 512-bit sẽ được xử lý qua 64 vòng lặp, trong mỗi vòng sẽ thực hiện các phép toán như XOR, cộng modulo, chuyển vòng và các phép toán khác để làm rối loạn dữ liệu.

Cập nhật giá trị băm

Sau khi tất cả các khối được xử lý, kết quả cuối cùng sẽ là giá trị băm của toàn bộ thông điệp ban đầu.

Chuẩn hóa thông điệp (Padding)

Bước 1: Chuyển đổi sang nhị phân

Chuyển đổi chuỗi đầu vào hex "6A09E667 BB67AE85 3C6EF372 A54FF53A 510E527F 9B05688C 1F83D9AB 5BE0CD19" sang dạng nhị phân: "0110 1010 ... 1001". và chuỗi này được chia ra thành các nhóm 32 bit được đánh kí tự lần lượt là A B C D E F G H

Bước 2: Thêm bit 1

Thêm một bit 1 vào cuối chuỗi nhị phân ban đầu. Chuỗi sau khi thêm bit 1 sẽ có độ dài là 257 bit.

Bước 3: Thêm bit 0

Thêm đủ các bit 0 sao cho độ dài của thông điệp sau khi thêm 1 và 0 là 448 bit modulo 512. Trong trường hợp này, cần phải thêm 256 bit 0 vào sau bit 1.

Bước 4: Thêm chiều dài ban đầu

Cuối cùng, thêm 64 bit đại diện cho chiều dài ban đầu của thông điệp (256 bit). Chiều dài 256 bit sẽ được chuyển thành dạng nhị phân là : 0000 0000 ... 0001 0000 0000.



Chia thành các khối nhỏ - Khởi tạo giá trị hằng số

Chia thông điệp thành các khối 512-bit

Vì thông điệp đã có độ dài là 512 bit, không cần phải chia thêm thông điệp thành nhiều khối. Chỉ có 1 khối 512-bit duy nhất để băm.

Khởi tạo giá trị băm ban đầu

SHA-256 sử dụng 8 giá trị khởi tạo sau cho các biến băm tạm thời:

H0 = 0x6A09E667

H1 = 0xBB67AE85

H2 = 0x3C6EF372

H3 = 0xA54FF53A

H4 = 0x510E527F

H5 = 0x9B05688C

H6 = 0x1F83D9AB

H7 = 0x5BE0CD19

Mở rộng các từ 32-bit

Mỗi khối 512-bit sẽ được chia thành 16 từ 32-bit [0 , 1 , 2 , . . . , 15] .

Trong trường hợp này, vì chúng ta chỉ có 1 khối, chúng ta sẽ có:

W0 = 0x6A09E667

W1 = 0xBB67AE85

W2 = 0x3C6EF372

W3 = 0xA54FF53A

W4 = 0x510E527F

W5 = 0x9B05688C

W6 = 0x1F83D9AB

W7 = 0x5BE0CD19

W8 → W15 = 0x00000000

Mở rộng các từ thành 64 từ

Dựa trên công thức mở rộng SHA-256:

$W_t = W(t - 16) + \Sigma_1(W(t-15)) + W(t - 7) + \Sigma_0(W(t-2))$

Tính toán các giá trị W16 đến W63 bằng cách áp dụng các phép toán Σ_1 và Σ_0 (xoay vòng và XOR) vào các từ trước đó.

$W_{16} = W_0 + \Sigma_1(W_1) + W_9 + \Sigma_0(W_{14}) = 0x6A09E667 + 0x67AE85BB + 0x00000000 + 0xB6BFFF7 = 0xC0C63B2E$



Made with Gamma

Vòng lặp tính toán

Sau khi mở rộng, thuật toán SHA-256 thực hiện **64 vòng tính toán** để cập nhật các giá trị băm tạm thời. Mỗi vòng lặp bao gồm các bước sau:

Bước 1: Tính toán T1 và T2

Trong mỗi vòng, tính toán các giá trị **T1** và **T2** như sau:

- $T1 = H + \Sigma_1(E) + Ch(E,F,G) + Kt + Wt$
- $T2 = \Sigma_0(A) + Maj(A,B,C)$

Các hàm và hằng số

- $\Sigma_1 = (E>>>6) \oplus (E>>>11) \oplus (E>>>25)$
- $\Sigma_0 = (A>>>2) \oplus (A>>>13) \oplus (A>>>22)$
- $Kt = \lfloor (\sqrt{2})^{\frac{t}{10000}} \rfloor$ (Ví dụ: $K0 = 0x428A2F98$, $K1 = 0x71374491$, ...)

Bước 2: Cập nhật giá trị băm

Cập nhật các giá trị băm tạm thời sau mỗi vòng:

- $H = G$
- $G = F$
- $F = E$
- $E = D + T1$
- $D = C$
- $C = B$
- $B = A$
- $A = T1 + T2$



Made with Gamma

Cập nhật giá trị băm toàn cục

Sau khi hoàn thành vòng lặp cho khối đầu tiên, các giá trị tạm thời A, B, C, D, E, F, G, H sẽ được cộng dồn vào các giá trị băm toàn cục. Bạn sẽ cộng các giá trị sau mỗi khối băm:

$$H0 \text{ (final)} = H0 \text{ (previous)} + A$$

$$H2 \text{ (final)} = H2 \text{ (previous)} + C$$

$$H4 \text{ (final)} = H4 \text{ (previous)} + E$$

$$H6 \text{ (final)} = H6 \text{ (previous)} + G$$

$$H1 \text{ (final)} = H1 \text{ (previous)} + B$$

$$H3 \text{ (final)} = H3 \text{ (previous)} + D$$

$$H5 \text{ (final)} = H5 \text{ (previous)} + F$$

$$H7 \text{ (final)} = H7 \text{ (previous)} + H$$

Sau khi tất cả các vòng lặp hoàn tất, sẽ có 256 bit cuối cùng là giá trị băm kết quả của SHA-256. Giá trị này là kết quả của toàn bộ quá trình băm cho thông điệp đầu vào.

Ứng dụng của SHA-256



Bảo mật mật khẩu

SHA-256 thường được sử dụng trong việc bảo mật mật khẩu người dùng. Thay vì lưu trữ mật khẩu gốc, hệ thống lưu trữ giá trị băm của mật khẩu. Khi người dùng đăng nhập, mật khẩu nhập vào được băm và so sánh với giá trị băm đã lưu trữ.



Chữ ký số

Trong các hệ thống xác thực, SHA-256 có thể được sử dụng để tạo chữ ký số, đảm bảo tính toàn vẹn của tài liệu hoặc giao dịch.



Blockchain

SHA-256 là nền tảng của các thuật toán khai thác và bảo mật trong blockchain. Ví dụ, trong Bitcoin, SHA-256 được sử dụng để bảo mật các khối giao dịch và xác thực chuỗi khối.



Kiểm tra toàn vẹn dữ liệu

SHA-256 giúp đảm bảo rằng dữ liệu không bị thay đổi trong quá trình truyền tải hoặc lưu trữ.



Ưu điểm và nhược điểm của SHA-256

Ưu điểm

- An toàn: Đến thời điểm hiện tại, SHA-256 vẫn được xem là rất an toàn và chưa bị phá vỡ bởi bất kỳ cuộc tấn công nào có hiệu quả đáng kể.
- Tiêu chuẩn quốc tế: SHA-256 được sử dụng rộng rãi và là một tiêu chuẩn mật mã mạnh mẽ trong nhiều ứng dụng bảo mật.

Nhược điểm

- Tốc độ: SHA-256 có thể không nhanh như các thuật toán băm khác như MD5 hay SHA-1, vì độ phức tạp tính toán của nó khá cao.
- Không thể đảo ngược: Mặc dù tính bảo mật rất cao, nhưng trong một số trường hợp, người ta cần phải có khả năng đảo ngược quá trình băm (mà SHA-256 không thể làm được).

Các Module Của Thuật Toán SHA-256

Khi thiết kế phần cứng cho thuật toán băm SHA-256, nó thường được chia thành các module con để cải thiện hiệu quả và tăng tính tối ưu hóa. Dưới đây là các module chính mà một thiết kế phần cứng có thể chia nhỏ trong việc thực thi SHA-256:

Module Chuẩn hóa (Preprocessing/Message Padding)

Chức năng: Trước khi tiến hành băm, dữ liệu cần được chuẩn hóa để đảm bảo rằng nó có độ dài là bội số của 512 bit. Quá trình này bao gồm:

- Thêm một bit 1 vào cuối thông điệp.
- Thêm các bit 0 cho đến khi độ dài thông điệp đạt độ dài cần thiết.
- Cuối cùng, bổ sung chiều dài của thông điệp gốc vào cuối thông điệp (64-bit).

Module Chia khối (Message Scheduling)

Chức năng: Sau khi chuẩn hóa, thông điệp sẽ được chia thành các khối 512-bit. Mỗi khối sẽ được chia tiếp thành các từ 32-bit.

Module Tính toán dữ liệu ban đầu (Calculate initial data)

Chức năng: Sau khi khối được chia thành 16 từ 32-bit, các từ này sẽ được mở rộng (expand) thành 64 từ. Các từ này được tạo ra thông qua một loạt các phép toán XOR, AND, OR và các phép toán số học khác. Quá trình này rất quan trọng trong việc làm thay đổi và khuếch đại dữ liệu.

Các Module Của Thuật Toán SHA-256

Module Tính toán Hash (Hash Calculation)

Chức năng: Đây là giai đoạn chính của thuật toán SHA-256, nơi thực hiện vòng lặp 64 lần để tính toán giá trị băm. Mỗi vòng lặp có các phép toán khác nhau với giá trị băm tạm thời (working variables) và giá trị mở rộng của các từ.

Các phép toán chính trong module này bao gồm:

- Chuyển động vòng (rotations)
- Phép toán XOR, AND, OR
- Cộng modulo
- Sử dụng các hằng số K trong SHA-256 (một tập hợp các số đã được xác định sẵn).

Module Cập nhật giá trị băm (Hash Value Update)

Chức năng: Sau mỗi vòng lặp, giá trị tạm thời của băm được cập nhật, và cuối cùng giá trị băm cuối cùng của khối 512-bit sẽ được cộng vào giá trị băm chung (các hằng số bắt đầu). Các giá trị này sẽ được lưu lại để tiếp tục quá trình băm cho các khối tiếp theo.

Module Điều khiển và Quản lý Dữ liệu (Control and Data Flow)

Chức năng: Đảm nhận vai trò điều phối các hoạt động giữa các module khác nhau. Chức năng này đảm bảo rằng dữ liệu được chuyển giao đúng cách qua các bước của thuật toán và đồng bộ hóa các phép toán giữa các module.

Các IP cần thiết cho hệ thống SoC của SHA-256

Để tích hợp thuật toán SHA-256 vào một hệ thống SoC (System on Chip), cần các IP sau:

Bộ xử lý trung tâm (CPU Core IP)

- Nếu SHA-256 được triển khai bằng phần mềm, cần một bộ xử lý nhúng mạnh mẽ như:
- ARM Cortex-A/M (cho hệ thống nhúng, IoT, mobile).
- RISC-V Core (mở nguồn, linh hoạt).
- Xtensa LX7 (dùng trong ESP32).
- Custom ASIC DSP (cho xử lý băm tốc độ cao).

Bộ nhớ và bộ điều khiển bộ nhớ (Memory Controller IP)

- SRAM/DRAM/NVM Controller IP để lưu trữ dữ liệu đầu vào/đầu ra của SHA-256.
- DMA Controller IP giúp truyền tải dữ liệu nhanh mà không cần CPU can thiệp.
- Cache Controller nếu CPU cần tối ưu hóa truy xuất dữ liệu.

Bộ điều khiển giao tiếp bus (Bus Interface IP)

- Cần IP kết nối giữa bộ xử lý và bộ gia tốc SHA-256:
- AMBA AXI/APB (nếu dùng ARM-based SoC).
- Wishbone/TileLink (nếu dùng RISC-V).
- AHB-Lite (cho hệ thống đơn giản).
- Bộ điều khiển bus giúp truyền dữ liệu giữa CPU, bộ nhớ và gia tốc SHA-256 một cách hiệu quả.



Các IP cần thiết cho hệ thống SoC của SHA-256

Để tích hợp thuật toán SHA-256 vào một hệ thống SoC (System on Chip), cần các IP sau:

Giao diện I/O (Peripheral IPs)

- Nếu SHA-256 dùng để mã hóa dữ liệu truyền qua mạng hoặc ngoại vi, cần:
 - UART/I2C/SPI Controller IP (giao tiếp thiết bị bên ngoài).
 - Ethernet MAC IP (bảo mật dữ liệu truyền qua mạng).
 - USB Controller IP (mã hóa dữ liệu truyền qua USB).

Bộ gia tốc mật mã SHA-256 (Cryptographic Accelerator IP)

- Nếu SHA-256 được triển khai bằng phần cứng, cần một SHA-256 Hardware Accelerator IP.
- Các IP này có thể là:
 - ARM CryptoCell (tích hợp trong chip ARM).
 - Open-source SHA-256 cores (VD: OpenCores, Xilinx IP).
 - Custom Verilog/VHDL SHA-256 Accelerator (tự thiết kế).
- Tính năng cần có:
 - Pipeline processing: Xử lý song song nhiều khối dữ liệu SHA-256.
 - DMA support: Truy xuất dữ liệu nhanh từ bộ nhớ mà không cần CPU.
 - Low-latency hashing: Tối ưu độ trễ tính toán.



Cảm ơn bạn đã lắng nghe!