## Hàm băm SHA

### Định nghĩa

SHA đƣợc thiết kế dựa trên những nguyên tắc của MD4/MD, tạo ra 160-bit giá trị Băm.

A, Miêu tả SHA

Cũng giống nhƣ với MD5, bức điện đƣợc cộng thêm một bit 1 và các bit 0 ở cuối bức điện để bức điện có thể chia hết cho 512. SHA sử dụng 5 thanh ghi dịch:

A = 0x67452301

B = 0xefcdab89 C = 0x98badcfe D = 0x10325476

E = 0xc3d2e1f0

Bức điện đƣợc chia ra thành nhiều khối 512-bit. Ta cũng đặt là a, b, c, d, e thay cho A, B, C, D, E đối với khối 512-bit đầu tiên của bức điện. SHA có bốn vòng lặp chính trong 5 giá trị a, b, c, d và e, sau đó cũng đƣợc cộng và dịch nhƣ trong MD5.

SHA xác lập bốn hàm phi tuyến nhƣ sau:

ft (X, Y, Z) = (X ∧ Y) V ((¬X) ∧ Z) với 0 <= t <= 19 ft (X, Y, Z) = X Y Z với 20 <= t <= 39

ft (X, Y, Z) = (X ∧ Y) V (X ∧ Z) V (Y ∧ Z) với 40 <=t <= 59 ft (X, Y, Z) = X Y Z với 60 <= t <=79.

Bốn hằng số sử dụng trong thuật toán là:

Kt = 21/2/4 = 0x5a827999 với 0 <= t <= 19

Kt = 31/2/4 = 0x6ed9eba1 với 20 <= t <=39 Kt = 51/2/4 = 0x8f1bbcdc vƣới 40 <=t <=59 Kt = 101/2/4 = 0xca62c1d6 với 60 <=t <= 79.

Các khối bức điện đƣợc mở rộng từ 16word đến 32-bit (M0 đến M15) thành 80 word 32-bit (W0 đến W79) bằng việc sử dụng thuật toán mở rộng:

Wt = Mt với 0 <= t <= 15

Wt = (Wt-3 Wt-8 Wt-16) với 16 <= t <= 79. Ta có thể miêu tả một vòng lặp của SHA nhƣ sau:

Diagram, schematic

Description automatically generated

*Sơ đồ một vòng lặp của SHA*

Nếu gọi Wt là biểu diễn của khối con thứ t của bức điện đƣợc mở rộng, và <<<s là biểu diễn dịch trái s bit, thì vòng lặp chính của SHA nhƣ sau:

a = A, b = B, c = C, d = D, e = E,

for t = 0 to 79

{

TEMP = (a <<<5) + ft(b, c, d) + e + Wt + Kb e = d,

d = c,

c = b <<<30,

b = a,

a = TEMP,

}

A = A + a, B = B + b, C = C + c, D = D + d, E = E + e,

Thuật toán tiếp tục với khối 512-bit tiếp theo cho tới khi hết bức điện, và kết quả sau cùng trong 4 thanh ghi A, B, C, D và E chính là hàm Băm SHA 160-bit.

1. Tính bảo mật trong SHA:

Để hiểu rõ hơn về tính bảo mật của SHA, ta hãy so sánh SHA với MD5 để có thể tìm ra những điểm khác nhau của hai hàm Băm này:

* + MD5 và SHA đều cộng thêm các bit “giả” để tạo thành những khối chia hết cho 512- bit, nhƣng SHA sử dụng cùng một hàm phi tuyến f cho cả bốn vòng.
  + MD5 sử dụng mỗi hằng số duy nhất cho mỗi bƣớc biến đổi, SHA sử dụng mỗi hằng số cho mỗi vòng biến đổi, hằng số dịch này là một số nguyên tố đối với độ lớn của word (giống với MD4).
  + Trong hàm phi tuyến thứ 2 của MD5 có sự cải tiến so với MD4, SHA thì sử dụng lại hàm phi tuyến của MD4, tức (X ∧ Y) V (X ∧ Z) V (Y ∧ Z).
  + Trong MD5 với mỗi bƣớc đƣợc cộng kết quả của bƣớc trƣớc đó. Sự khác biệt đối với SHA là cột thứ 5 đƣợc cộng (không phải b, c hay d nhƣ trong MD5), điều này làm cho phƣơng pháp tấn công của Boer-Bosselaers đối với SHA bị thất bại (den Boer và Bosselaers là hai ngƣời đã phá thành công 2 vòng cuối trong MD4).

Cho đến nay, chƣa có công bố nào đƣợc đƣa ra trong việc tấn công SHA, bởi vì độ dài của hàm Băm SHA là 160-bit, nó có thể chống lại phƣơng pháp tấn công bằng vét cạn (kể cả Birthday attack) tốt hơn so với hàm Băm MD5 128-bit.

### Ứng dụng của hàm băm

Ứng dụng chính của hàm Băm là sử dụng với hệ chữ ký điện tử, trong đó thay vì ký trực tiếp lên các văn bản, thông điệp (mà trong đa số trƣờng hợp là rất lớn, tốc độ chậm) ngƣời ta sẽ ký lên giá trị băm đại diện cho toàn bộ văn bản đó. Điều này đặc biệt quan trọng và hiệu quả bởi vì chúng ta biết rằng các hệ chữ ký điện tử đều làm việc với các phép tính số học số lớn nên bản thân chúng đã tƣơng đối chậm, việc sử sụng giá trị băm thay cho toàn bộ văn bản là giải pháp toàn diện khắc phục đƣợc yếu điểm này của các hệ chữ ký điện tử. Ngoài việc sử dụng với các hệ chữ ký điện tử hàm Băm còn đƣợc sử dụng vào các mục đích khác nhƣ: xác thực hóa thông điệp, xác thực hóa ngƣời dùng.

Đối với các ứng dụng không cần giữ bí mật thông điệp mà chỉ cần đảm bảo thông điệp không bị thay đổi trên đƣờng truyên ngƣời ta sẽ sử dụng hàm băm cho mục đích xác thực tính nguyên vẹn của thông điệp đó. Chẳng hạn chúng ta có một phần mềm mã nguồn mở ở dạng setup muốn phân phối cho ngƣời dùng, rõ ràng việc gửi phầm mềm đó tới máy tính của ngƣời dùng là không cần phải mã hóa, tuy nhiên nếu nhƣ phần mềm đó (khi đó phần mềm chính là thông điệp). Ngƣời dùng sẽ download cả phần mềm và giá trị băm nhận đƣợc, sau đó tiền hành băm lại, đối sánh hai giá trị này nếu khớp nhau thì có thể đảm bảo phần mềm không bị sửa đổi trên đƣờng truyền. Hiện nay đa số các phần mềm mã nguồn mở đều đƣợc phân phối theo cách này.

Trong các hệ thống yêu cầu có xác thực ngƣời dùng nhƣ các hệ quản trị cơ sở dữ liệu, hệ điều hành, các ứng dụng web, ứng dụng dạng desktop application, để lƣu mật khẩu ngƣời dùng ngƣời ta cũng sử dụng các hàm Băm hoặc các hệ mã trong các vai trò của hàm Băm (không sử dụng khóa). Khi đó mỗi tài khoản của ngƣời dùng thay vì lƣu dƣới dạng tên truy cập (user name) và mật khẩu (password) sẽ đƣợc lƣu dƣới dạng: tên ngƣời dùng, giá trị băm của mật khẩu. Khi một ngƣời dùng đăng nhập vào hệ thống, hệ thống sẽ lấy tên truy cập, mật khẩu họ nhập vào, kiểm tra xem có tên truy cập nào nhƣ vậy hay không. Nếu có sẽ tiến hành băm giá trị mật khẩu do ngƣời dùng nhập vào.

# CHƢƠNG 3. KIẾN THỨC LĨNH HỘI VÀ BÀI HỌC KINH NGHIỆM

## Nội dung đã thực hiện

* + 1. ***Nội dung học được thông qua thực hiện bài tập lớn***
       - Khái niệm về chữ ký điện tử
       - Hệ chữ ký ElGamal
       - Chuẩn chữ ký điện tử
       - Mô hình ứng dụng của chữ ký điện tử
       - Mã hóa bất đối xứng.
       - Cách hoạt động của mã hóa bất đối xứng
       - Ứng dụng của mã hóa bất đối xứng
       - Lợi ích và hạn chế của mã hóa bất đối xứng
       - So sánh mã hóa bất đối xứng và mã hóa đối xứng
       - Đặc tính của hàm băm
       - Hàm băm SHA-1 và SHA-2
       - Ứng dụng của hàm băm
       - Bài toán chữ ký điện tử ElGamal
       - Nắm vững một ngôn ngữ lập trình cơ bản (Java, C#, C++, Python) và giải đƣợc bài toán để ứng dụng vào thực tiễn
    2. ***Bài học kinh nghiệm được rút ra sau khi kết thúc bài tập lớn***

Trong quá trình xây dựng và phát triển phần mềm, nhóm đã gặp rất nhiều khó khăn trong việc xây dựng phần mềm do còn nhiều hạn chế về kiến thức hệ thống cũng nhƣ kiến thức về lĩnh vực hƣớng tới của phần mềm.

Xong hệ phần mềm đã đáp ứng các nhu cầu cơ bản. Nhƣng cần bổ sung thêm một số tính năng, chỉnh sửa giao diện sao cho đẹp mắt, tối ƣu nhất.

Chƣơng trình đƣợc kiểm tra rà soát sao cho việc xuất hiện lỗi là tối thiểu, ít nhất. Đảm bảo chƣơng trình hoạt động một cách mƣợt mà nhất. Tối ƣu giao diện và đơn giản hóa giúp cho việc sử dụng dễ dàng hơn.

Mục tiêu đạt đƣợc: Hoàn thiện một phần mềm với những chức năng cơ bản, cần thiết nhất thuận tiện cho ngƣời dùng. Hạn chế lỗi ở mức tối thiểu.

Giao diện tối giản hóa, dễ sử dụng.

Chƣa đạt đƣợc: Giao diện cổ điển, chƣa bắt mắt, ít chức năng.

Kết thúc việc xây dựng phần mềm, nhóm đã đúc kết đƣợc nhiều kinh nghiệm, học hỏi thêm đƣợc nhiều kiến thức trong việc phát triển phần mềm, góp phần hoàn thiện kỹ năng, tuy duy trong xây dựng phần mềm và củng cố kiến thức đã học.