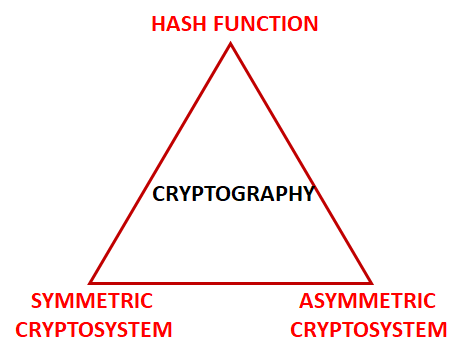
**2 MÃ HÓA ỨNG DỤNG **

**MÃ HÓA THÔNG TIN**

Mật mã học (*cryptology*) là ngành khoa học nghiên cứu các phương pháp liên quan đên bảo mật thông tin và truyền thông qua việc sử dụng các kỹ thuật biến đổi thông tin. Một cách tổng quát, mật mã học bao gồm (**i**) mã hóa (*cryptography*) nghiên cứu cách xây dựng các hệ mã và (**ii**) thám mã (*cryptoanalysis*) nghiên cứu cách phân tích các hệ mã đã được công bố:

CRYPTOLOGY = CRYPTOGRAPHY + CRYPTOANALYSIS

**Ví dụ mở đầu**

Bằng cách sử dụng bảng ký hiệu chuẩn của Hoa Kỳ về trao đổi thông tin – **ASCII** (American Standard Code for Information Interchange), với chữ ‘A’ được biểu diễn bằng 1 byte (8 bit nhị phân) có dạng 01000001 tương ứng với 65 trong hệ thập phân, ‘B’ là 66, …, ‘Z’ – 90. Và ta giả sử thêm dấu cách (khoảng trắng) có mã 64 (thực ra là ký hiệu của ‘@’), thì chuỗi

S = ‘KHOA CONG NGHE THONG TIN’ ứng với

M = 757279656467797871647871726964847279787164847378.

Dòng số trên chỉ là cách mã hóa theo quy ước (*encode*) không phải là mật mã (*encrypt*) vì mọi người có thể lấy từng cặp số trong dãy trừ đi 64 sẽ được lại dòng chữ gốc (*decode*).

***Ví dụ 1***: giả sử lấy từng cặp số **m** trong chuỗi số và biến đổi thành **c** theo công thức

c = Ek(m) = 64 + (m – 64 + k)%26

Lấy bí mật **k** = 3, ta được dãy số

C = 787582686770828174678174757267877582817467877681,

và tra ngược bảng mã ASCII, ta được chuỗi ký tự

W = NKRD FRQJ QJKH WKRQJ WLQ.

Ta thấy, chỉ những ai biết **k** = 3 mới đọc được ý nghĩa của chuỗi W.

***Ví dụ 2***: giả sử với **k** = 4018975632 và ta lấy từng chữ số **m** trong dãy số M biến đổi thành số **c** theo công thức

c = σ(m), với σ(0123456789) = k = 4018975632.

Từ chuỗi

M = 75727965 67797871 78717269 8472797871 847378.

ta được chuỗi

C = 67616257 56626360 63606852 3961626360 396863

***Ví dụ 3***: tương tự, ta có thể dùng k = σ(0123456789) = 0185947263.

***Ví dụ 4***: lấy kết quả của ví dụ 3, biểu diễn lại bằng cách gom 8 số lại thành nhóm và cộng các nhóm lai. Kết quả được bao nhiêu chia cho p = 7919, giữ lại phần dư, ta được

h = H(C) = (67616257 + 56626360 + 63606852 + 39616263 + 60396863) % 7919 = 6945.

Rõ ràng, từ h = 6945, rất khó để biết nó được tính từ C.

**Định nghĩa hình thức**

Các ví dụ trên cho ta ý niệm ban đầu về các hệ mật mã (*cryptosystem*) mà là một ánh xạ (*mapping*) biến đổi từng thành phần của chuỗi ký tự cần mã hóa sau khi đã số hóa (*digitalize*), được gọi là thông điệp (*message*) hay bản rõ (*plain text*) thành các thành phần của dãy số mật, gọi là bản mã (*cipher text*) bằng cách kết hợp với một thông tin mật, gọi là khóa (*key*), để chỉ người có khóa tương ứng mới có thể hiểu được thông điệp gốc. Một cách hình thức, ta định nghĩa một hệ mã như sau:

Một hệ mã là một ánh xạ Ek “khả nghịch” đi từ không gian các thông điệp vào không gian các bản mã với k là một khóa trong không gian khóa :

, sao cho

.

Hàm Ek(m) gọi là hàm mã hóa (encryption), mã hóa thông điệp m thành bản mã c, sử dụng khóa mã hóa k (*encryption key*). Và hàm Dk’(c) là hàm giải mã (decryption) mà giải mã bản mã c thành lại bản rõ m, sử dụng khóa giải mã k’ (*decryption key*).

Tùy thuộc độ phức tạp của hàm mã hóa, hoặc tùy thuộc quan hệ giữa khóa mã hóa và khóa giải mã, hoặc có sử dụng khóa hay không mà ta có thể phân biệt các hệ mã khác nhau.

**Phân loại các hệ mã**

***Bí mật hoàn toàn – Hàm mã hóa là ánh xạ đơn giản***

Như trong ví dụ 1, cho dù có bí mật khóa k thì chỉ bẳng cách thử (vét cạn) qua 26 trường hợp, bản rõ có thể được phục hồi từ bản mã được cho. Trong trường hợp nay, phải bí mật cả thuật toán (hàm mã hóa) lẫn khóa mã hóa.

***Công bố thuật toán và bí mật thông tin khóa – k ≡ k’***

Các ví dụ 2 và ví dụ 3 minh họa hệ mã này. Trong các hệ mã này, khóa bí mật là dãy số, mà là hoán vị của 10 chữ số, và hàm mã hóa chỉ là thay thế thông điệp m bằng hoán vị tương ứng σ(m) của nó. Trong ví dụ này, không gian phải vét cạn là 10! = 3628800 khả năng. Nếu hoán vị của 100 ký hiệu thì khả năng vét cạn 100! Trường hợp sẽ là không khả thi. Hệ mã dạng này được gọi là mã đối xứng (*symmetric cryptosystem*) khi khóa giải mã có thể được suy ra dễ ràng từ khóa mã hóa. Khóa mã hóa là bí mật chung giữa 2 đối tác trong hệ thống, vì thế phải được giữ bí mật, hệ mã có tính chất này còn được gọi là mã (khóa) bí mật (*secret key cryptosystem*). Rõ ràng khóa phải được quy ước trước giữa 2 đối tác, ta cũng có thể gọi là hệ mã quy ước khóa trước (*pre-share key cryptosystem*).

***Công bố thuật toán và một phần thông tin khóa – k ≠ k’***

k trong ví dụ 3 có tính chất đáng lưu ý là y = σ(x) = x3 % 11 và σ-1(y) = y11 % 11 = x. Như vậy, nếu ký hiêu e = k = 3 là khóa mã hóa thì ta có thể dùng d = k’ = 7 để giải mã. Theo đó, mã hóa bằng một khóa, khóa mã hóa e (*encryption key*), và giải bằng khóa khác, khóa giải mã (*decryption key*). Như vậy, sử dụng hệ mã này ta có thể công khai khoa mã hóa e, gọi là khóa công khai (*public key*), và giữ bí mật khóa giải mã d, gọi là khóa cá nhân (*private key*). Nên hệ mã dạng này được gọi là hệ mã (khóa) công khai (*public key cryptosystem*) hay còn có tên khác là hệ mã bất đối xứng (*asymmetric cryptosystem*) do tính khác biệt của các khóa mã và giải mã.

***Không sử dụng khóa – k = ‘’***

Ví dụ 4 minh họa cách hoạt động của hệ mã không sử dụng khóa. Hệ mã này đơn giản là biến đổi chuỗi thông điệp dài bất kỳ thành một chuỗi có chiều dài cố định. Hàm biến đổi E(.) này, giờ được ký hiệu là H(.), được gọi là hàm băm mật mã, và giá trị h = H(S) băm một chuỗi S được gọi là trị băm. Hàm này đơn giản chỉ là giấu thông tin (gốc) và thương được sử dụng trong các mục đích xác thực.

Các mạng xâu khối sử dụng chính hai loại hệ mã: mã công khai và hàm băm. Hệ mã bất đối xứng được sử dụng với mục đích bảo mật thông tin và không phải phổ biến.

**Nguyên lý thiết kế các hệ mã**

Về hình thức, ta thấy, để xây dựng một hệ mã, đơn giản chỉ là xác định 3 không gian: không gian các thông điệp (), không gian các bản mã (), không gian khóa (), và thiết kế thuật toán cho hàm khả nghịch E. Tuy nhiên, do yêu cầu bảo mật, các hệ mã phải được thiết kể để đảm bảo an toàn, bảo mật và hiệu quả khi dùng. Trong thực tế, các hệ mã thường được thiết kế theo nguyên lý Kerchkoff.

***Nguyên lý Kerchkoff***

. Hệ mã được chứng minh không thể bị thám mã về mặt toán học (*analyzing*), hơn nữa, phải không thể bị phá vỡ trong thực tế (*hacking*).

. Hệ mã nếu có bị kẻ xâm nhập khống chế cũng sẽ không bị bất kỳ sự xâm phạm nào vào hệ thống, và không gây bất kỳ sự bất tiện nào cho người dùng.

. Khóa mã hóa phải dễ trao đổi, dễ nhớ và có thể thay đổi.

. Bản mã có thể được truyền trên các kênh không an toàn.

. Thiết bị mã hóa và các văn bản phải có tính động và có thể sử dụng với người.

. Cuối cùng, hệ mã cần dễ sử dụng, không đòi hỏi phải có kiến thức và các quy tắc phức tạp phải tuân thủ.

**Về tính bảo mật**

Một hệ mã, trước hết phải bảo mật và kháng được mọi tấn công hệ thống (*hacking*) cũng như phân tích thuật toán (*analyzing*).

***Tấn công hệ mã***

Các cuộc tấn công thường được phân loại theo hành động được kẻ tấn công thực hiện. Theo đó, tấn công có thể là thụ động hoặc tấn công chủ động

*Tấn công thụ động*. Mục tiêu chính là giành được quyền truy cập (trái phép) vào thông tin.

*Tấn công tích cực*. Một cuộc tấn công tích cực liên quan đến việc thay đổi thông tin theo một cách nào đó qua việc tiến hành một số quy trình trên thông tin.

***Các giải định kẻ tấn công có thể có***

*Môi trường xung quanh hệ thống mật mã*. Kẻ tấn công có thể thâm nhập vào môi trường mà các thiết bị hay hệ mã đang hoạt động.

*Chi tiết về sơ đồ mã hóa*. Kẻ tấn công có được chi tiết về các lược đồ, các giao thức mật mã.

*Sự sẵn có của bản mã*. Kẻ tấn công giả thiết thu thập được không giới hạn các bản mã.

*Sự sẵn có của bản rõ và bản mã*. Kẻ tấn công được giải thiết thu thập được tập (vô hạn) các cặp bản rõ và bản mã.

Từ các giả định trên, các kỹ thuật tấn công có thể được tiến hành phù hợp.

***Phân loại tân công***

*Tấn công chỉ dựa vào bản mã*: **COA** – **Cipher text Only Attack**

Tấn công chỉ dựa vào bản mã (COA) là mô hình thám mã khi kẻ tấn công chỉ có một tập các bản mã. Mặc dù chỉ có các bản mã nhưng trong thực tế, kẻ tấn công có thể có một số tri thức về bản rõ. Chẳng hạn, kẻ tấn công có thể biết ngôn ngữ của bản rõ (tiếng Việt, tiếng Anh, …) hoặc/và biết được phân phối các ký tự trong bản rõ. Cũng vậy, dữ liệu cũng như các thông điệp trong các giao thức, trong nhiều hệ thống, thường là một phần của bản rõ và có thể đoán hoặc biết trước.

Tấn công thành công khi phục hồi được bản rõ tương ứng (tấn công một phần), hoặc phục hồi đươc khóa mã hóa (tấn công toàn phần). Thậm chí, thu thập được bất kỳ thông tin nào về bản rõ ngoài những gì kẻ tấn công đã biết vẫn được coi là thành công. Chẳng hạn, nhiều giao thức gửi bản mã liên tục nhằm duy trì trạng thái bảo mật đường truyền, việc phân biệt các tin nhắn thực với các tin nhắn rỗng có thể xem là thành công, thậm chí dự đoán được về sự tồn tại của một số thông điệp thực cũng sẽ tạo điều kiện thuận lợi cho việc phân tích lưu lượng truy cập.

*Tấn công biết bản rõ:* **KPA** *–* **Known Plaintext Attack**

Tấn công biết bản đã biết (KPA) là mô hình thám mã trong đó kẻ tấn công thu thập được tập các bản rõ và bản mã tương ứng. Tập dữ liệu nay có thể được sử dụng để phục hồi một số thông tin bí mật như khóa cá nhân bí mật và mã từ điển (code book).

*Tấn công chọn bản rõ*: **CPA – Chosen Plaintext Attack**

Tấn công chọn bản rõ (CPA) là mô hình thám mã, giả định rằng kẻ tấn công có thể thu thập được các bản mã của các bản gốc hắn muốn. Mục tiêu của tấn công này là lấy thông tin làm giảm tính bảo mật của sơ đồ mã hóa.

*Tấn công từ điển* – **Dictionary Attack**

Tấn công từ điển dựa vào việc thử tất cả các chuỗi trong danh sách được sắp xếp trước. Tấn công từ điển có thể sử dụng các danh sách có sẵn trên Internet chứa hàng trăm triệu mật khẩu được khôi phục từ các vụ vi phạm dữ liệu đã xảy ra. Ngoài ra có thể sử dụng phần mềm bẻ khóa áp dụng vào các danh sách đó để tạo ra các biến thể khác, như thay thế các chữ số cho các chữ cái. Tấn công từ điển thường thành công vì nhiều người có xu hướng chọn mật khẩu ngắn và là những từ thông thường hoặc mật khẩu thông dụng. Tấn công từ điển cũng có thể thành công vì nhiều kỹ thuật tạo mật khẩu thông dụng đều nằm trong danh sách có sẵn. Kết hợp với việc tạo mẫu của phần mềm bẻ khóa, khả năng thành công càng cao.

*Tấn công vét cạn*: **BFA – Brute Force Attack**

Tấn công vét cạn là tấn công mà kẻ tấn công kiểm tra một cách có hệ thống tất cả mật khẩu và cụm mật khẩu có thể có cho đến khi tìm thấy mật khẩu chính xác. Ngoài ra, kẻ tấn công có thể thử đoán khóa được tạo từ mật khẩu bằng cách sử dụng hàm phái sinh khóa.

Trên lý thuyết, tấn công vét cạn là tấn công thám mã có thể được áp dụng để thử giải mã bất kỳ bản mã nào. Tấn công như vậy được sử dụng khi không thể tận dụng các điểm yếu khác trong hệ thống mã hóa (nếu có).

Khi đoán mật khẩu, tấn công vét cạn hiệu quả khi kiểm tra tất cả mật khẩu ngắn, đối với mật khẩu dài, các phương pháp khác như tấn công từ điển hiệu quả hơn. Mật khẩu, cụm mật khẩu và khóa dài khó bị bẻ khóa hơn so với những mật khẩu ngắn hơn.

Tấn công vét cạn sẽ không hiệu quả với cách làm xáo trộn dữ liệu mã hóa khiến kẻ tấn công khó nhận ra khi nào mã đã bị bẻ khóa hoặc chí ít cũng khiến kẻ tấn công phải làm nhiều việc hơn để kiểm tra từng lần đoán. Tấn công vét cạn cũng được dùng để đánh giá độ án toàn của một hệ thống.

Tấn công vét cạn thực chất là áp dụng của tìm kiếm vét cạn. Kỹ thuật chung là liệt kê tất cả các ứng cử viên và kiểm tra từng ứng cử viên.

*Tấn công sinh nhật* – **Birth Attack**

Tấn công sinh nhật là một kiểu tấn công bằng cách khai thác tính chất toán học của bài toán sinh nhật trong lý thuyết xác suất. Tấn công sinh nhật dựa trên nguyên lý chuồng câu, theo đó, số chim nhiêu hơn số chuồng thì chắc chắn có chuồng có nhiều hơn 1 con, nghĩa là có đụng độ. Tấn công sinh nhật được dùng để tìm ra xung đột trong hàm băm. Có giả thuyết là máy tính lượng tử có thể thực hiện các cuộc tấn công sinh nhật hiệu quả. Tuy nhiên, giải thuyết này còn nhiều tranh cãi.

*Tân công qua trung gian*: **MIMA** – **Man in the Middle Attack**

Tấn công qua trung gian (MIM) là tấn công mạng trong đó kẻ tấn công bí mật chuyển tiếp và có thể thay đổi thông tin liên lạc giữa đối tác đang liên lạc mà họ không nhận ra có kẻ đứng giữa họ. Kẻ tấn công phải có khả năng chặn tất cả các tin nhắn truyền giữa hai nạn nhân và tiêm những tin nhắn mới vào. Việc này trong nhiều trường hợp có thể thực hiện dễ dàng. Chẳng hạn, trong khu vực phát song wifi không mã hóa, kẻ tấn công có thể phá vỡ tiến trình xác thực lẫn nhau.

*Tấn công kênh phụ*: **SCA – Side Chanel Attack**

Tấn công kênh phụ là tấn công nào trên thông tin bổ sung có thể được thu thập do cách thức cơ bản mà giao thức hoặc thuật toán máy tính được triển khai, không phải do sai sót trong thiết kế giao thức hoặc thuật toán hoặc do những sơ suất nhỏ nhưng có khả năng gây thiệt hại nghiêm trọng trong quá trình triển khai. Thông tin về thời gian, mức tiêu thụ điện năng, rò rỉ điện từ và âm thanh là những ví dụ về thông tin bổ sung có thể bị khai thác để tạo điều kiện cho các cuộc tấn công kênh phụ. Tấn công kênh phụ thường yêu cầu kiến thức kỹ thuật về hoạt động của hệ thống

*Tấn công định thời* – **Timming Attack**

Tấn công định thời là một dạng của tấn công kênh phụ trong đó kẻ tấn công thâm nhập hệ thống mật mã và phân tích thời gian thực hiện các thuật toán mật mã. Mọi thao tác logic trong máy tính đều cần có thời gian thực thi và thời gian này dài ngắn khác nhau phụ thuộc đầu vào; nếu đo chính xác được thời gian các thao tác, kẻ tấn công có thể suy ngược lại đầu vào. Việc phục hồi bí mật qua thông tin thời gian có thể dễ hơn nhiều việc sử dụng phương pháp thám mã đã biết ở trên. Đôi khi thông tin về thời gian cũng được kết hợp với thám mã để tăng tốc độ phục hồi thông tin.

Thông tin có thể bị rò rỉ qua việc đo thời gian cần thiết để phản hồi các truy vấn cụ thể. Tấn công theo thời gian có thể được áp dụng cho bất kỳ thuật toán nào có sự thay đổi thời gian phụ thuộc vào dữ liệu. Việc loại bỏ sự phụ thuộc vào thời gian là khó khăn trong một số thuật toán sử dụng các chỉ thị cấp thấp thường có thời gian thực hiện khác nhau.

Tấn công theo thời gian thường bị bỏ qua trong giai đoạn thiết kế vì chúng phụ thuộc quá nhiều vào việc triển khai và có thể được đưa vào một cách vô ý khi tối ưu hóa trình biên dịch. Vì thế, cần lưu ý tấn công khai thác thời gian thực thin gay trong quá trình thiết kế thuật toán, giao thức bảo mật.

*Tấn công phân tích năng lượng* – **Power Analysis Attacks**

Phân tích năng lượng là một hình thức tấn công kênh phụ trong đó kẻ tấn công nghiên cứu mức tiêu thụ năng lượng của thiết bị phần cứng mật mã. Tấn công này dựa vào các đặc tính vật lý cơ bản của thiết bị. Bằng cách đo dòng điện, có thể biết được một lượng nhỏ thông tin về dữ liệu đang được xử lý.

Phân tích năng lượng đơn giản (SPA – Simple Power Analysis) liên quan đến việc phân tích số liệu thống kê năng lượng hoặc biểu đồ hoạt động dòng điện theo thời gian. Phân tích năng lượng vi phân (DPA – Diferential Power Analysis) là một hình thức phân tích năng lượng hiện đại, cho phép kẻ tấn công tính toán các giá trị trung gian trong tính toán mật mã thông qua phân tích thống kê dữ liệu được thu thập từ nhiều hoạt động mật mã.

*Tấn công phân tích lỗi* – **Fault analysis Attacks**

Phân tích lỗi vi sai (DFA – Differential Fault Analysis) là một dạng tấn công kênh phụ vào các hê mã, cụ thể là thám mã. Nguyên tắc là tạo ra các lỗi cho các hoạt động mã hóa để tiết lộ trạng thái bên trong của chúng.

**HỆ MÃ KHÓA CÔNG KHAI**

**Mã khóa công khai**

Hệ mã khóa công khai, hay mã bất đối xứng, là các mã sử dụng cặp khóa có quan hệ toán học với nhau. Mỗi cặp khóa gồm một khóa công khai (*public key*) và một khóa cá nhân (*private key*). Các cặp khóa được tạo bằng thuật toán mật mã dựa trên một bài toán khó. Bài toán khó được sử dụng để tạo ra hàm một chiều. Tính bảo mật của mật mã khóa công khai phụ thuộc vào việc giữ bí mật khóa cá nhân; khóa công khai có thể được công bố mà không ảnh hưởng đến bảo mật của hệ mã.

Trong hệ mã khóa công khai, bất kỳ ai có khóa công khai đều có thể mã hóa tin nhắn, tạo ra bản mã, nhưng chỉ những người biết khóa cá nhân tương ứng mới có thể giải mã để phục hồi lại bản rõ.

**Chữ ký số**

Trong hệ thống chữ ký số, người gửi sử dụng khóa cá nhân cùng với thông điệp để tạo ra chữ ký. Bất kỳ ai có khóa công khai tương ứng đều có thể xác minh xem chữ ký có khớp với thông điệp hay không. Người giả mạo không biết khóa riêng sẽ không thể tìm ra bất kỳ cặp thông điệp/chữ ký nào vượt qua quá trình xác minh bằng khóa công khai.

**Ứng dụng của mã công khai**

. Ứng dụng rõ ràng nhất của hệ mã công khai là mã hóa thông tin liên lạc để cung cấp tính bảo mật - một tin nhắn mà người gửi mã hóa bằng khóa công khai của người nhận, chỉ có thể được giải mã bằng khóa nhân tương ứng của người nhận.

. Một ứng dụng khác của mã công khai là chữ ký số. Lược đồ chữ ký số có thể được sử dụng để xác thực người gửi. Hệ thống chống từ chối sử dụng chữ ký số để đảm quyền tác giả của một tài liệu hoặc thông tin liên lạc.

. Các ứng dụng khác được xây dựng trên nền tảng mã công khai có thể kể đến: tiền kỹ thuật số (cryptocurrency), trao đổi/thiết lập khóa cho các hệ mã đối xứng, chứng thực, và các dịch vụ đánh dấu thời gian và các giao thức chống từ chối.

**Điểm yếu của các hệ mã công khai**

Như mọi hệ thống liên quan đến bảo mật, điều quan trọng là xác định các điểm yếu tiềm ẩn. Ngoài việc lựa chọn thuật toán khóa bất đối xứng yếu hoặc độ dài khóa quá ngắn, rủi ro bảo mật chính là khóa cá nhân bị lộ.

Ngoài ra, với sự ra đời của điện toán lượng tử, nhiều thuật toán khóa bất đối xứng được coi là dễ bị tấn công và các kế hoạch kháng lượng tử mới đang được phát triển để khắc phục vấn đề này.

**HÀM BĂM MẬT MÃ**

**Hàm băm**

Hàm băm mật mã (**CHF** – Cryptographic Hash Function) là một thuật toán băm (ánh xạ của chuỗi nhị phân bất kỳ thành chuỗi nhị phân có kích thước cố định là n bit và có các tính chất:

. Xác suất tìm ra một chuỗi từ một trị băm n-bit cụ thể là 2-n (đối với bất kỳ hàm băm tốt nào). Do đó giá trị băm có thể được sử dụng làm đại diện cho thông báo;

. Kháng đụng độ một. Việc tìm chuỗi đầu vào khớp với giá trị băm cho trước (tiền ảnh) là bất khả thi.

. Kháng đụng độ hai. TVieecj tìm rah ai chuỗi có cùng trị băm là không thể

Hàm băm mật mã có nhiều ứng dụng bảo mật thông tin, đặc biệt là trong chữ ký số, mã xác thực thông điệp (MAC – *Message Authentication Code*) và các hình thức xác thực khác. Chúng cũng có thể được sử dụng để lập chỉ mục dữ liệu trong bảng băm, để lấy dấu vân tay, để phát hiện dữ liệu trùng lặp hoặc nhận dạng tính duy nhất các tập tin và kiểm tổng (*checksum*) để phát hiện lỗi dữ liệu. Trị băm mật mã đôi khi được gọi là dấu vân (*fingerprint*), tổng kiểm tra hoặc đơn giản là trị băm, cho thấy các ứng dụng có thể có của hàm băm.

Khac với hàm băm mật mã, hàm băm phi mật mã được sử dụng trong các bảng băm và để phát hiện các lỗi vô tình, việc xây dựng chúng thường không có khả năng chống lại các tấn công có chủ ý. Ví dụ: tấn công từ chối dịch vụ vào các bảng băm có thể xảy ra nếu các xung đột được tìm thấy dễ dàng.

**Tính chất**

Hàm băm mật mã thường được thiết kế để nhận đầu vào là một chuỗi có độ dài và tạo ra giá trị băm có độ dài cố định.

Hàm băm mật mã phải có khả năng chống lại tất cả các kiểu tấn công phân tích mật mã đã biết. Trong mật mã lý thuyết, mức độ bảo mật của hàm băm mật mã đã được xác định bằng các thuộc tính sau:

***Kháng tiền ảnh một***

Cho trước trị băm h, sẽ khó tìm được bất kỳ thông điệp m nào sao cho h = hash(m). Khái niệm này có liên quan đến hàm một chiều. Các hàm không có tính chất này có thể bị tấn công tiền ảnh.

***Kháng đụng độ yếu***

Với đầu vào m1, khó tìm được đầu vào m2 ≠ m1 sao cho hash(m1) = hash(m2). Tính chất này còn được gọi là khả năng kháng đụng độ yếu. Các hàm không có tính chất này dễ bị tấn công tiền ảnh loại hai.

***Kháng đụng độ mạnh***

Rất khó để tìm được hai thông điệp khác nhau m1 và m2 sao cho hash(m1) = hash(m2). Cặp thông điệp như vậy được gọi là độ độ. Tính chất này còn được gọi là khả năng kháng đụng độ mạnh. Trị băm cần dài ít nhất gấp đôi giá trị cần thiết cho khả năng kháng tiền ảnh; nếu không thì đụng độ có thể được tìm ra bằng tấn công sinh nhật.

Khả năng kháng đụng độ ngụ ý khả năng kháng tiền ảnh hai nhưng không bao hàm khả năng kháng tiền ảnh.

Các tính chất này có nghĩa là kẻ xấu không thể thay thế hoặc sửa dữ liệu đầu vào mà không thay đổi trị băm, gọi là bản tóm tắt của văn bản đầu vào. Vì vậy, nếu hai chuỗi có cùng một bản tóm tắt, ta có thể kết luận chúng giống hệt nhau. Khả năng kháng tiền ảnh thứ hai ngăn kẻ tấn công tạo một văn bản có cùng trị băm với văn bản mà kẻ tấn công không biết. Khả năng kháng đụng độ ngăn kẻ tấn công tạo hai văn bản khác nhau có cùng trị băm.

Một hàm đáp ứng các tiêu chí này vẫn có thể có những tính chất không mong muốn. Hiện tại, các hàm băm mật mã phổ biến rất dễ bị tấn công kéo dài độ dài: cho trước hash(m) và len(m) chứ không phải m, bằng cách chọn một m′ phù hợp, kẻ tấn công có thể tính toán hash(m ∥ m′), trong đó ∥ là phép ghép nối. Tính chất này có thể được sử dụng để phá vỡ các sơ đồ xác thực đơn giản dựa trên hàm băm. Việc tạo mã chứng thực thông điệp dùng hàm băm – HMAC (Hash Message Authentication Code) khắc phục được vấn đề này.

Trong thực tế, khả năng kháng đụng độ không đủ cho nhiều mục đích sử dụng thực tế. Ngoài khả năng kháng đụng độ, kẻ tấn công không thể tìm thấy hai thông điệp có nội dung cơ bản giống nhau; hoặc suy ra được bất kỳ thông tin hữu ích nào về dữ liệu, mà chỉ dựa trên bản tóm tắt của dữ liệu. Cụ thể, hàm băm phải hoạt động giống hàm ngẫu nhiên nhất có thể (thường được gọi là sấm ngữ ngẫu nhiên (oracle) trong các bằng chứng về bảo mật) trong khi vẫn có tính xác định và tính toán hiệu quả. Điều này loại trừ các hàm như hàm băm xác suất dựa trên biên đổi Fourier– SWIFFT, có thể được chứng minh chặt chẽ là có khả năng kháng đụng độ với giả định rằng một số bài toán là khó tính toán, nhưng, với một hàm tuyến tính, không thỏa mãn các tính chất bổ sung này.