# **RESEARCH**

**(Mã hóa công khai và chữ ký số)**

**Member :**

Hà Tuấn Giang

Nguyễn Công Tú

Hoàng Thị Ngọc

Nguyễn Đức Toàn

# **I,Mã Hóa Công Khai**

## **1.Nói qua về lịch sử hình thành của khoa học mật mã**

## Mật mã học là gì ? là một lĩnh vực liên quan đến các kỹ thuật ngôn ngữ và toán học để đảm bảo [an toàn thông tin](https://vi.wikipedia.org/wiki/An_to%C3%A0n_th%C3%B4ng_tin), cụ thể là trong [thông tin liên lạc](https://vi.wikipedia.org/wiki/Th%C3%B4ng_tin_li%C3%AAn_l%E1%BA%A1c). Trong lịch sử, mật mã học gắn liền với quá trình [*mã hóa*](https://vi.wikipedia.org/wiki/M%C3%A3_h%C3%B3a); điều này có nghĩa là nó gắn với các cách thức để chuyển đổi [thông tin](https://vi.wikipedia.org/wiki/Th%C3%B4ng_tin) từ dạng này sang dạng khác nhưng ở đây là từ dạng thông tin có thể nhận thức được thành dạng không nhận thức được, làm cho thông tin trở thành dạng không thể đọc được nếu như không có các kiến thức bí mật.

## Lịch sử hình thành: Mật mã học có lịch sử lâu dài và đầy màu sắc. Nói chung, những dạng sớm nhất của cách viết bí mật (ngày nay gọi chung là *mật mã hóa cổ điển*) chỉ cần có bút và giấy. Hai phạm trù chính của mật mã cổ điển là [mật mã hoán vị](https://vi.wikipedia.org/w/index.php?title=M%E1%BA%ADt_m%C3%A3_ho%C3%A1n_v%E1%BB%8B&action=edit&redlink=1), trong đó người ta sắp xếp lại trật tự các chữ cái của thông điệp, và [mật mã thay thế](https://vi.wikipedia.org/w/index.php?title=M%E1%BA%ADt_m%C3%A3_thay_th%E1%BA%BF&action=edit&redlink=1), trong đó người ta thay thế có hệ thống các chữ cái hay các nhóm chữ cái bằng các chữ cái hay các nhóm chữ cái khác. Văn bản được mật mã hóa bằng mật mã cổ điển có xu hướng lộ ra các thông tin thống kê nhất định về văn bản thường. Bằng cách sử dụng các thông tin này, mật mã cổ điển rất dễ bị dò ra (ví dụ bằng [phân tích tần suất](https://vi.wikipedia.org/wiki/Ph%C3%A2n_t%C3%ADch_t%E1%BA%A7n_su%E1%BA%A5t)). Mật mã cổ điển vẫn còn được phổ biến tới ngày nay, chủ yếu thông qua việc giải các [ô đố chữ](https://vi.wikipedia.org/w/index.php?title=%C3%94_%C4%91%E1%BB%91_ch%E1%BB%AF&action=edit&redlink=1) (xem [tài liệu viết bằng mật mã](https://vi.wikipedia.org/w/index.php?title=T%C3%A0i_li%E1%BB%87u_vi%E1%BA%BFt_b%E1%BA%B1ng_m%E1%BA%ADt_m%C3%A3&action=edit&redlink=1)).

## Các thiết bị và các kỹ thuật khác nhau đã được sử dụng để mật mã hóa. Một trong những thiết bị sớm nhất có lẽ là [gậy mật mã](https://vi.wikipedia.org/wiki/G%E1%BA%ADy_m%E1%BA%ADt_m%C3%A3) ([tiếng Hy Lạp](https://vi.wikipedia.org/wiki/Ti%E1%BA%BFng_Hy_L%E1%BA%A1p): σκυτάλη). Trong nửa đầu thế kỷ XX, một số thiết bị cơ khí đã được phát minh để thực hiện mật mã hóa, bao gồm [rotor machines](https://vi.wikipedia.org/w/index.php?title=Rotor_machine&action=edit&redlink=1) — nổi tiếng nhất là [máy Enigma](https://vi.wikipedia.org/wiki/M%C3%A1y_Enigma) được người Đức sử dụng trong [Đại chiến thế giới 2](https://vi.wikipedia.org/wiki/Chi%E1%BA%BFn_tranh_th%E1%BA%BF_gi%E1%BB%9Bi_th%E1%BB%A9_hai). Mật mã thực hiện bằng các máy móc này đã tăng độ phức tạp lên đáng kể đối với công việc phân tích mã.

## Với sự ra đời của máy tính kỹ thuật số và [điện tử học](https://vi.wikipedia.org/wiki/%C4%90i%E1%BB%87n_t%E1%BB%AD_h%E1%BB%8Dc) thì các mật mã cực kỳ phức tạp đã có thể được thực hiện. Đặc trưng của mật mã máy tính là chúng thực hiện trên các chuỗi [nhị phân](https://vi.wikipedia.org/wiki/H%E1%BB%87_nh%E1%BB%8B_ph%C3%A2n), không giống như trong các mô hình mật mã hóa cổ điển và cơ học (chỉ sử dụng bảng chữ cái với khoảng 26 ký tự-phụ thuộc vào từng ngôn ngữ). Mật mã máy tính cũng có khả năng chịu đựng việc phân tích mật mã tốt hơn; rất ít các mật mã như thế dễ bị tổn thương chỉ bởi kiểu tấn công biết bản mã.

## Các nghiên cứu rộng rãi có tính học thuật về mật mã hóa hiện đại là tương đối gần đây — nó chỉ được bắt đầu trong cộng đồng mở kể từ những năm thập niên 1970 với các chi tiết kỹ thuật của [DES](https://vi.wikipedia.org/wiki/DES_(%C4%91%E1%BB%8Bnh_h%C6%B0%E1%BB%9Bng)) (viết tắt trong tiếng Anh của *Data Encryption Standard* tức *Tiêu chuẩn Mật mã hóa Dữ liệu*) và sự phát minh ra [RSA](https://vi.wikipedia.org/wiki/RSA_(m%C3%A3_h%C3%B3a)). Kể từ đó, mật mã hóa đã trở thành công cụ được sử dụng rộng rãi trong liên lạc và bảo mật máy tính.

## Cũng giống như các bài học thu được từ trong lịch sử của nó, các nhà mật mã hóa cũng rất thận trọng khi nhắc đến tương lai. [Định luật Moore](https://vi.wikipedia.org/wiki/%C4%90%E1%BB%8Bnh_lu%E1%BA%ADt_Moore) thông thường được nhắc đến khi nói về [độ lớn khóa](https://vi.wikipedia.org/wiki/%C4%90%E1%BB%99_l%E1%BB%9Bn_kh%C3%B3a), và các hiệu ứng tiềm năng của [máy tính lượng tử](https://vi.wikipedia.org/wiki/M%C3%A1y_t%C3%ADnh_l%C6%B0%E1%BB%A3ng_t%E1%BB%AD) cũng đã được nói

## 

## **2.Tổng quát về mã hóa công khai**

### 2.0.Chức năng cơ bản của mật mã

**Tính bí mật (confidentiality)**: nó đảm bảo tính bí mật của dữ liệu mà mình gửi đi và chỉ những người liên quan mới biết được nội dung

· **Tính toàn vẹn (integrity)** : đảm bảo dữ liệu không thể bị mất mát hoặc chỉnh sửa trong qua trình gửi và nhận mà không bị phát hiện.

· **Tính xác thực (authenticity)**: đảm bảo danh tính của thực thể được xác minh.

· **Tính không thể chối từ**: đảm bảo người gửi không thể chối cãi với thông tin mình gửi đi

## 

### 2.1. Mã hóa bí mật ( mã hóa đối xứng )

### Nguyên lý:

* key mã hóa + key giải mã là giống nhau.

Vd: sơ khai nhất là mật mã Caesar : một dạng mật mã thay thế -à Mỗi chữ cái sẽ được thay thế bằng chữ cái sau nó 3 đơn vị trong bảng chữ cái, A → D, B → E

### Hạn chế:

. Do dùng chung key --> cần có kênh đứng giữa trao đổi key trong lần đầu tiên có thể sẽ không an toàn.

. Tốn nhiều khóa trong để đảm bảo bí mật trong một nhóm 20 người : n(n-1)/2

. Khó ứng dụng trong các hệ thống mở

### 2.2 Mã hóa công khai (mã hóa bất đối xứng)

Như đã thấy ở trên, mã hóa bí mật gặp rất nhiều hạn chế trong việc trao đổi mã, tính an toàn không được đảm bảo.

Mã hóa công khai ra đời đã cải thiện được rất nhiều điểm yếu của mã hóa bí mật.

### Nguyên lý:

* Gồm: 1 khóa công khai + 1 khóa bí mật
* Hệ mật bao gồm :

. Bản rõ (plaintext-M): bản tin được sinh ra bởi bên gửi

. Bản mật (ciphertext-C): bản tin che giấu thông tin của bản rõ, được gửi tới bên nhận qua một kênh không bí mật

. Khóa: Bên nhận có 1 cặp khóa:

. Khóa công khai (Kub) : công bố cho tất cả mọi người biết (kể cả hacker)

. Khóa riêng (Krb) : bên nhận giữ bí mật, không chia sẻ cho bất kỳ ai

* Mã hóa (encrypt-E): C = E(Kub, M)
* Giải mã (decrypt): M = D(Krb, C) = D(Krb, E(Kub, M))

+ *Yêu cầu đối với cặp khóa (Kub, Krb) là:*

* Hoàn toàn ngẫu nhiên
* Có quan hệ về mặt toán học 1-1.
* Nếu chỉ có giá trị của Kub không thể tính được Krb.
* Krb phải được giữ bí mật hoàn toàn.

### Ưu điểm :

* Không cần chia sẻ khóa mã hóa một cách bí mật ( vì nó sẽ được công khai )
* Khóa giải mã chỉ có người nhận biết ( Bob )
* Cần ít cặp khóa hơn mã hóa công khai : n cặp khóa cho n người.

### Nhược điểm :

Tuy có nhiều ưu điểm khắc phục được hạn chế của mã hóa bí mật, mã hóa công khai cũng có những nhược điểm dễ nhìn thấy. Để đảm bảo được độ bí mật, mã hóa công khai sẽ dùng số n càng lớn dẫn đến số lượng tính toán lớn. Vì vậy, trong thực tế, học thường dùng mã hóa công khai kết hợp với mã hóa bí mật.

## 

### Tính an toàn

Về khía cạnh an toàn, các thuật toán [mã hóa bất đối xứng](https://vi.wikipedia.org/w/index.php?title=M%E1%BA%ADt_m%C3%A3_h%C3%B3a_kh%C3%B3a_b%E1%BA%A5t_%C4%91%E1%BB%91i_x%E1%BB%A9ng&action=edit&redlink=1) cũng không khác nhiều với các thuật toán [mã hóa đối xứng](https://vi.wikipedia.org/wiki/Thu%E1%BA%ADt_to%C3%A1n_kh%C3%B3a_%C4%91%E1%BB%91i_x%E1%BB%A9ng). Có những thuật toán được dùng rộng rãi, có thuật toán chủ yếu trên lý thuyết; có thuật toán vẫn được xem là an toàn, có thuật toán đã bị phá vỡ... Cũng cần lưu ý là những thuật toán được dùng rộng rãi không phải lúc nào cũng đảm bảo an toàn.

Một số thuật toán có những chứng minh về độ an toàn với những tiêu chuẩn khác nhau.

Nhìn chung, chưa có thuật toán nào được chứng minh là an toàn tuyệt đối (như hệ thống [mật mã sử dụng một lần](https://vi.wikipedia.org/w/index.php?title=M%E1%BA%ADt_m%C3%A3_s%E1%BB%AD_d%E1%BB%A5ng_m%E1%BB%99t_l%E1%BA%A7n&action=edit&redlink=1)). Vì vậy, cũng giống như tất cả các thuật toán mật mã nói chung, các thuật toán mã hóa khóa công khai cần phải được sử dụng một cách thận trọng.

### Các ứng dụng

* Loại mật mã này được sử dụng bởi nhiều hệ thống máy tính hiện đại nhằm cung cấp bảo mật cho thông tin nhạy cảm: email,...
* Giao thức Lớp cổng bảo mật (SSL) giúp kết nối an toàn đến các trang web có thể cũng sử dụng mật mã bất đối xứng
* Ngoài ra, nó còn góp mặt nổi bật trong công nghệ blockchain và tiền điện tử

### 2.3 Những vấn đề đặt ra trong thực tế

### Sự tương thích với bưu chính

Để thấy rõ hơn ưu điểm của hệ thống [mã hóa khóa bất đối xứng](https://vi.wikipedia.org/w/index.php?title=M%E1%BA%ADt_m%C3%A3_h%C3%B3a_kh%C3%B3a_b%E1%BA%A5t_%C4%91%E1%BB%91i_x%E1%BB%A9ng&action=edit&redlink=1) ta có thể dùng sự tương thích với hệ thống bưu chính trong ví dụ sau:

2 người (Alice và Bob) trao đổi thông tin mật thông qua hệ thống bưu chính. Alice cần gửi một bức thư có nội dung cần giữ bí mật tới cho Bob và sau đó nhận lại thư trả lời (cũng cần giữ bí mật) từ Bob.

**Trong hệ thống** [**mã hóa khóa đối xứng**](https://vi.wikipedia.org/wiki/Thu%E1%BA%ADt_to%C3%A1n_kh%C3%B3a_%C4%91%E1%BB%91i_x%E1%BB%A9ng), Alice sẽ cho bức thư vào hộp và khóa lại rồi gửi hộp theo đường bưu chính bình thường tới cho Bob. Khi Bob nhận được hộp, anh ta dùng một khóa giống hệt như khóa Alice đã dùng để mở hộp, đọc thông tin và gửi thư trả lời theo cách tương tự. Vấn đề đặt ra là Alice và Bob phải có 2 khóa giống hệt nhau bằng một cách an toàn nào đó từ trước (chẳng hạn như gặp mặt trực tiếp).

**Trong hệ thống** [**mật mã hóa khóa bất đối xứng**](https://vi.wikipedia.org/w/index.php?title=M%E1%BA%ADt_m%C3%A3_h%C3%B3a_kh%C3%B3a_b%E1%BA%A5t_%C4%91%E1%BB%91i_x%E1%BB%A9ng&action=edit&redlink=1), Bob và Alice có hai khóa khác nhau. Alice sử dụng khóa nhận được từ Bob để khóa hộp. Khi nhận được hộp đã khóa bằng khóa công khai của mình, Bob có thể mở khóa và đọc thông tin. Để trả lời Alice, Bob cũng thực hiện theo quá trình tương tự với khóa của Alice.

Điều quan trọng nhất ở đây là Bob và Alice không cần phải gửi đi khóa bí mật của mình. Điều này làm giảm nguy cơ một kẻ thứ 3 (chẳng hạn như một nhân viên bưu chính biến chất) làm giả khóa trong quá trình vận chuyển và đọc những thông tin trao đổi giữa 2 người trong tương lai. Thêm vào đó, trong trường hợp Bob do sơ suất làm lộ khóa của mình thì các thông tin do Alice gửi cho người khác vẫn giữ bí mật (vì sử dụng các cặp khóa khác).

### Những điểm yếu

Tồn tại khả năng một người nào đó có thể tìm ra được khóa bí mật. Không giống với hệ thống [mật mã sử dụng một lần](https://vi.wikipedia.org/w/index.php?title=M%E1%BA%ADt_m%C3%A3_s%E1%BB%AD_d%E1%BB%A5ng_m%E1%BB%99t_l%E1%BA%A7n&action=edit&redlink=1) (one-time pad) hoặc tương đương, chưa có thuật toán mã hóa khóa bất đối xứng nào được [chứng minh](https://vi.wikipedia.org/wiki/Ch%E1%BB%A9ng_minh) là an toàn trước các tấn công dựa trên bản chất [toán học](https://vi.wikipedia.org/wiki/To%C3%A1n_h%E1%BB%8Dc) của thuật toán.

Khả năng một mối quan hệ nào đó giữa 2 khóa hay điểm yếu của thuật toán dẫn tới cho phép giải mã không cần tới khóa hay chỉ cần khóa mã hóa vẫn chưa được loại trừ. An toàn của các thuật toán này đều dựa trên các ước lượng về [khối lượng tính toán](https://vi.wikipedia.org/w/index.php?title=Kh%E1%BB%91i_l%C6%B0%E1%BB%A3ng_t%C3%ADnh_to%C3%A1n&action=edit&redlink=1) để giải các bài toán gắn với chúng. Các ước lượng này lại luôn thay đổi tùy thuộc khả năng của [máy tính](https://vi.wikipedia.org/wiki/M%C3%A1y_t%C3%ADnh) và các [phát hiện](https://vi.wikipedia.org/wiki/Ph%C3%A1t_minh) [toán học](https://vi.wikipedia.org/wiki/To%C3%A1n_h%E1%BB%8Dc) mới.

Mặc dù vậy, độ an toàn của các thuật toán mật mã hóa khóa công khai cũng tương đối đảm bảo. Nếu thời gian để phá một mã (bằng [phương pháp duyệt toàn bộ](https://vi.wikipedia.org/w/index.php?title=Ph%C6%B0%C6%A1ng_ph%C3%A1p_duy%E1%BB%87t_to%C3%A0n_b%E1%BB%99&action=edit&redlink=1)) được ước lượng là 1000 năm thì [thuật toán](https://vi.wikipedia.org/wiki/Thu%E1%BA%ADt_to%C3%A1n) này hoàn toàn có thể dùng để [mã hóa](https://vi.wikipedia.org/wiki/M%C3%A3_h%C3%B3a) các [thông tin](https://vi.wikipedia.org/wiki/Th%C3%B4ng_tin) về [thẻ tín dụng](https://vi.wikipedia.org/wiki/Th%E1%BA%BB_t%C3%ADn_d%E1%BB%A5ng) - Rõ ràng là thời gian phá mã lớn hơn nhiều lần thời gian tồn tại của thẻ (vài năm).

Một điểm yếu tiềm tàng trong việc sử dụng khóa bất đối xứng là khả năng bị tấn công dạng man in the middle attack: kẻ tấn công lợi dụng việc phân phối khóa công khai để thay đổi khóa công khai. Sau khi đã giả mạo được khóa công khai, kẻ tấn công đứng ở giữa 2 bên để nhận các [gói tin](https://vi.wikipedia.org/w/index.php?title=G%C3%B3i_tin&action=edit&redlink=1), [giải mã](https://vi.wikipedia.org/w/index.php?title=Gi%E1%BA%A3i_m%C3%A3&action=edit&redlink=1) rồi lại [mã hóa](https://vi.wikipedia.org/wiki/M%C3%A3_h%C3%B3a) với khóa đúng và gửi đến nơi nhận để tránh bị phát hiện. Dạng tấn công kiểu này có thể phòng ngừa bằng các phương pháp [trao đổi khóa](https://vi.wikipedia.org/w/index.php?title=Trao_%C4%91%E1%BB%95i_kh%C3%B3a&action=edit&redlink=1) an toàn nhằm đảm bảo [nhận thực](https://vi.wikipedia.org/w/index.php?title=Nh%E1%BA%ADn_th%E1%BB%B1c&action=edit&redlink=1) người gửi và [toàn vẹn](https://vi.wikipedia.org/w/index.php?title=To%C3%A0n_v%E1%BA%B9n&action=edit&redlink=1) thông tin. Một điều cần lưu ý là khi các chính phủ quan tâm đến dạng tấn công này: họ có thể thuyết phục (hay bắt buộc) nhà [cung cấp chứng thực số](https://vi.wikipedia.org/w/index.php?title=Cung_c%E1%BA%A5p_ch%E1%BB%A9ng_th%E1%BB%B1c_s%E1%BB%91&action=edit&redlink=1) xác nhận một khóa giả mạo và có thể đọc các thông tin mã hóa.

### Khối lượng tính toán

Để đạt được độ an toàn tương đương, [thuật toán](https://vi.wikipedia.org/wiki/Thu%E1%BA%ADt_to%C3%A1n) [mã hóa khóa bất đối xứng](https://vi.wikipedia.org/w/index.php?title=M%E1%BA%ADt_m%C3%A3_h%C3%B3a_kh%C3%B3a_b%E1%BA%A5t_%C4%91%E1%BB%91i_x%E1%BB%A9ng&action=edit&redlink=1) đòi hỏi khối lượng tính toán nhiều hơn đáng kể so với thuật toán [mã hóa khóa đối xứng](https://vi.wikipedia.org/wiki/Thu%E1%BA%ADt_to%C3%A1n_kh%C3%B3a_%C4%91%E1%BB%91i_x%E1%BB%A9ng). Vì thế trong thực tế hai dạng thuật toán này thường được dùng bổ sung cho nhau để đạt hiệu quả cao. Trong mô hình này, một bên tham gia trao đổi thông tin tạo ra một khóa đối xứng dùng cho phiên giao dịch. Khóa này sẽ được trao đổi an toàn thông qua hệ thống mã hóa khóa bất đối xứng. Sau đó 2 bên trao đổi thông tin bí mật bằng hệ thống mã hóa đối xứng trong suốt phiên giao dịch.

## 3.Các loại mã hóa công khai(giới thiệu khái quát)

-,Trao đổi khóa Diffie-Hellman: là một phương pháp trao đổi [khóa](https://vi.wikipedia.org/wiki/Kh%C3%B3a_(m%E1%BA%ADt_m%C3%A3_h%E1%BB%8Dc)) được phát minh sớm nhất trong [mật mã học](https://vi.wikipedia.org/wiki/M%E1%BA%ADt_m%C3%A3_h%E1%BB%8Dc). Phương pháp trao đổi khóa Diffie–Hellman cho phép hai bên (người, thực thể giao tiếp) thiết lập một khóa [bí mật chung](https://vi.wikipedia.org/w/index.php?title=B%C3%AD_m%E1%BA%ADt_chung&action=edit&redlink=1) để mã hóa dữ liệu sử dụng trên [kênh truyền thông](https://vi.wikipedia.org/w/index.php?title=K%C3%AAnh_(truy%E1%BB%81n_th%C3%B4ng)&action=edit&redlink=1) không an toàn mà không cần có sự thỏa thuận trước về khóa bí mật giữa hai bên. Khóa bí mật tạo ra sẽ được sử dụng để mã hóa dữ liệu với phương pháp mã hóa [khóa đối xứng](https://vi.wikipedia.org/wiki/Kh%C3%B3a_%C4%91%E1%BB%91i_x%E1%BB%A9ng).

-,Hệ thống mật mã Rabin: là một kỹ thuật [mật mã](https://wivi.wiki/wiki/Cryptographic) không đối xứng , mà tính bảo mật của nó, giống như [RSA](https://wivi.wiki/wiki/RSA_(algorithm)) , có liên quan đến độ khó của việc [phân tích nhân tử số nguyên](https://wivi.wiki/wiki/Integer_factorization) . Tuy nhiên, hệ thống mật mã Rabin có ưu điểm là nó đã được chứng minh về mặt toán học là an toàn về mặt tính toán trước một cuộc tấn công chọn bản rõ, miễn là kẻ tấn công không thể tính toán số nguyên một cách hiệu quả, trong khi không có bằng chứng nào được biết về RSA. [[1]](https://wivi.wiki/wiki/Rabin_cryptosystem#wiko) : 145 Nó có nhược điểm là mỗi đầu ra của hàm Rabin có thể được tạo ra bởi bất kỳ đầu vào nào trong số bốn đầu vào có thể có; nếu mỗi đầu ra là một bản mã, thì cần có thêm độ phức tạp khi giải mã để xác định đầu vào nào trong bốn đầu vào có thể là bản rõ thực sự.

-,Hệ thống mật mã Paillier(Thường được dùng

trong bỏ phiếu điện tử): là một sản phẩm của việc nghiên cứu Mã hóa đồng hình hoàn toàn (fully homomorphic encryption - FHE).

Mã hóa đồng hình hoàn toàn (FHE) là một hình thức mã hóa tiên tiến hơn đã được phát triển trong thập kỷ qua, được thiết kế để thu hẹp khoảng cách nói trên - bằng cách cho phép dữ liệu vẫn được mã hóa ngay cả trong quá trình tính toán.

Phương pháp toán học đằng sau FHE được thiết kế để các phép tính có thể thực hiện trực tiếp trên dữ liệu được mã hóa (ciphertext) mà không cần các dịch vụ đi kèm để theo dõi liệu việc mã hóa dữ liệu đó có cung cấp kết quả chính xác hay không.

-;mã hóa Paillier, sử dụng các hoạt động bổ sung, sẽ phù hợp nhất cho các ứng dụng liên quan đến bỏ phiếu vì nó cho phép người dùng thêm các giá trị khác nhau theo cách không thiên vị trong khi giữ giá trị riêng tư. Công nghệ này không chỉ có thể bảo vệ dữ liệu khỏi sự thao túng. Nó có thể cho phép xác minh độc lập bởi các bên thứ ba được ủy quyền.

-,EI-Gamal:Hệ mã hóa Elgamal là hệ mã hóa phi đối xứng. áp dụng bài toán Logarit rời rạc nên độ an toàn của nó phụ thuộc vào độ phức tạp của bài toán Logarit rời rạc

-,Mã hóa đường cong elliptic:**Mật mã đường cong elip (** **Elliptic Curve Cryptography**-**ECC)** sử dụng các tính chất toán học của đường cong elliptic để tạo ra các hệ thống mật mã khóa công khai. Giống như tất cả các mật mã khóa công khai, ECC dựa trên các hàm toán học đơn giản để tính toán theo một hướng, nhưng rất khó để đảo ngược. Trong trường hợp của ECC, khó khăn này nằm ở chỗ không khả thi khi tính toán lôgarit qrời rạc của phần tử đường cong elliptic ngẫu nhiên liên quan đến điểm cơ sở đã biết công khai, hoặc "bài toán lôgarit rời rạc đường cong elip" (ECDLP)[1](https://www.ssl.com/vi/faqs/M%E1%BA%ADt-m%C3%A3-%C4%91%C6%B0%E1%BB%9Dng-cong-elliptic-ecc-l%C3%A0-g%C3%AC/#1). Các **Thuật toán chữ ký số đường cong Elliptic (ECDSA)** là một thuật toán ký được sử dụng rộng rãi cho mật mã khóa công khai sử dụng ECC

-,RSA

## 4.EXAMPLE(phân tích thuật toán RSA)

**1.**  Lấy 2 snt: p , q

**2.** P x q = n

**3.** (p-1)(q-1) = φ (n)

- Định nghĩa: φ(n) : là số các số nguyên dương nhỏ hơn hoặc bằng n, nguyên tố cùng nhau với n.

UCNL ( n, a ) = 1 → φ(n) = tổng số nguyên dương a thỏa mãn

- Chứng minh công thức trên:

+ Dễ thấy : Nếu p là snt —> φ(p) = p-1 ( các số từ 1 —> p-1)

+ Ta thấy p & q nguyên tố cùng nhau .

Gọi số m là số thỏa mãn: UCLN(m,n) != 1

Vậy nên, m sẽ có dạng: m = px ( x : 1 → q ) —> Có q số m thỏa mãn

Hoặc : m = qx ( x: 1 → p ) —> Có p số m thỏa mãn

Vì : p, q là 2 số nguyên tố cùng nhau —> sẽ có 1 trường hợp m lặp lại giống nhau: m = pq = qp = n

—> Có : ( p + q – 1 ) số m thỏa mãn : UCLN (m,n) != 1

Vì vậy: φ(n) = n – (p+q-1)

= p1p2−(p1+p2−1)

= p1(p2−1)−(p2−1)

= (p1−1) x (p2−1)

= φ(p1) × φ(p2)

**4.** Lấy e nguyên tố cùng nhau vs " φ n " --> e = 2^16 + 1 = 65537

—> Bản chất có thể lấy luôn 1 snt , người ta mặc định lấy số e = 65537

**5**. Lấy d sao cho: (d.e) đồng dư vs 1 ( mod φ(n) ) ---> tức: (d.e) % φ(n) = 1

---> d và e nghịch đảo modun của nhau theo modun " φ(n) "

\* Định lý nghịch đảo module : Nếu số nguyên dương n và số nguyên a nguyên tố cùng nhau thì tồn tại duy nhất một số nguyên x thuộc (0,1,...,n-1)

—> Áp dụng vào trường hợp trên:

Số nguyên dương φ(n) và e nguyên tố cùng nhau —> Tồn số nguyên d thuộc (0,1,… φ(n)-1) thỏa mãn.

**6**. Thông điệp mã hóa: c = me mod n

—> n : công khai (public key)

**7**. Giải mã: r = cd mod n

—> d : bí mật (private key)

\* Giải thích :

- n : công khai

- e : cố định, ai cũng biết

- NGƯỜI NHẬN (BOB) : d là mã bí mật chỉ BOB biết

- n càng lớn ( nhiều bit ) , càng mạnh ---> k tốt cho các thiết bị điện thoại, máy tính bảng ( RAM ít )

- Mỗi người tự sinh ra 1 cặp chìa khóa công khai và bí mật

- Để phá mã RSA, hoàn toàn dựa vào kỹ thuật phân tích thừa số nguyên tố. Chưa có thuật toán nào làm được bằng thời gian đa thức (thời gian máy tính giải quyết được trong tgian con người chờ đợi được. )

- Sử dụng thực tế, số n dùng khoảng 2000 chữ số nhị phân —> Máy tính chưa thể tính toán được (mất cả trăm tỉ năm để giải ra)

**II. Hash funtion**

1. Hashing là gì

Hashing là quá trình biến đầu vào là một nội dung có kích thước, độ dài bất kỳ rồi sử dụng những thuật toán, công thức toán học để biến thành đầu ra tiêu chuẩn có độ dài nhất định. Quá trình đó sử dụng những Hàm băm (Hash function).

1. Hàm băm mật mã

Như tên gọi của nó vậy, những hàm băm như vậy được sử dụng vào mục đích mã hóa dữ liệu. Những hàm băm mật mã cũng giống như những hàm băm thông thường, nhưng mang trong mình một số đặc điểm khác, quan trọng nhất là không thể đảo ngược. Điều này có nghĩa là khi bạn có trong tay giá trị sau khi băm, bạn không thể biết giá trị ban đầu là gì. Điều này đơn giản như việc bạn uống một cốc sinh tố hỗn hợp rất ngon nhưng không thể nào liệt kê tất cả nguyên liệu của nó vậy. Ngoài ra, hàm băm 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)

1. Hash dùng để làm gì?

Kiểm tra sự toàn vẹn của tệp tin

Như đặc điểm của hàm băm, cùng một giá trị sẽ cho ra cùng một giá trị băm. Vậy nên ta có thể đối chiếu tệp tin ta tải trên mạng về với bản gốc bằng cách so sánh giá trị băm của chúng với nhau.

Xác minh mật khẩu

Có một điều rất hay bạn nên biết, trong những thiết kế cơ sở dữ liệu hiện đại, thứ lưu trong đó không phải mật khẩu của bạn dưới dạng văn bản đơn thuần mà là giá trị hash của chúng. Khi bạn nhập mật khẩu, mật khẩu của bạn sẽ được chạy qua hàm hash, sau đó sẽ được so sánh với giá trị băm trong cơ sở dữ liệu để quyết định bạn có được chứng thực để sử dụng dịch vụ không. Điều này làm giảm đáng kể thiệt hại khi cơ sở dữ liệu bị tấn công, khi những gì bị lộ ra ngoài là những giá trị băm chứ không phải mật khẩu của bạn

1. Tạm kết

Như các bạn có thể thấy, Hashing là một kỹ thuật rất quan trọng và có tính ứng dụng cực kỳ cao, đặc biệt trong lĩnh vực bảo mật. Những nhà phát triển liên tục phải thiết kế ra những thuật toán Hash mới, do những thuật toán cũ nhanh chóng bị giải mã bằng sự phát triến nhanh như vũ bão của công nghệ. Một khi một thuật toán nào đó bị crack (bẻ khóa), nó không còn an toàn tuyệt đối. Hiểu được những thuộc tính cơ bản và cách thức hoạt động chính của hàm băm là một việc không thể thiếu đối với mọi lập trình viên.

# **III. Digital signature**

**Tại sao phải sử dụng chữ ký số (Digital Signatures)???**

1. **Using Digital Signatures**

Chữ ký số là một kỹ thuật toán học được sử dụng để cung cấp tính xác thực, tính toàn vẹn và không từ chối. Chữ ký số có các thuộc tính cụ thể cho phép xác thực thực thể và toàn vẹn dữ liệu. Ngoài ra, chữ ký số cung cấp sự tính không chối bỏ của giao dịch. Nói cách khác, Chữ ký số đóng vai trò là bằng chứng pháp lý cho thấy việc trao đổi dữ liệu đã diễn ra. Chữ ký số sử dụng mật mã không đối xứng.

* Xác thực (Authentic): Chữ ký không thể bị giả mạo và cung cấp bằng chứng rằng người ký chứ không phải ai khác, đã ký vào tài liệu.
* Không thể thay đổi (Unalterable): Sau khi một tài liệu được ký, nó sẽ không thể thay đổi được.
* Không thể tái sử dụng (Not Reusable): Không thể chuyển chữ ký tài liệu sang tài liệu khác.
* Không chối bỏ (Non-repudiated): Tài liệu đã ký được coi là giống như một tài liệu vật lý. Chữ ký là bằng chứng cho thấy tài liệu đã được ký bởi người thực.

Chữ ký số thường được sử dụng trong hai trường hợp sau:

1. Ký mã (Code signing) - Điều này được sử dụng cho các mục đích xác thực và toàn vẹn dữ liệu. Việc ký mã được sử dụng để xác minh tính toàn vẹn của các tệp thực thi được tải xuống từ trang web của nhà cung cấp. Nó cũng sử dụng các chứng chỉ kỹ thuật số đã ký để xác thực và xác minh danh tính của trang web là nguồn của các tệp.

2. Chứng thư số (Digital certificates) - Những chứng chỉ này tương tự như một thẻ ID ảo và được sử dụng để xác thực danh tính của hệ thống với trang web của nhà cung cấp và thiết lập kết nối được mã hóa để trao đổi dữ liệu bí mật.

Có ba thuật toán Tiêu chuẩn chữ ký số (DSS) được sử dụng để tạo và xác minh chữ ký số:

• Thuật toán chữ ký số (DSA) - DSA là tiêu chuẩn ban đầu để tạo các cặp khóa công khai và khóa riêng, cũng như để tạo và xác minh chữ ký số.

• Thuật toán Rivest-Shamir Adelman (RSA) - RSA là một thuật toán bất đối xứng thường được sử dụng để tạo và xác minh chữ ký số.

• Thuật toán chữ ký số đường cong Elliptic (ECDSA) - ECDSA là một biến thể mới hơn của DSA và cung cấp tính năng xác thực và không thoái thác chữ ký số với các lợi ích bổ sung về hiệu quả tính toán, kích thước chữ ký nhỏ và băng thông tối thiểu.

1. **Digital Signatures for Code Signing**

Chữ ký số thường được sử dụng để đảm bảo tính xác thực và tính toàn vẹn của mã phần mềm. Các tệp thực thi được bao bọc trong một phong bì có chữ ký kỹ thuật số, cho phép người dùng cuối xác minh chữ ký trước khi cài đặt phần mềm.

Mã ký số cung cấp một số đảm bảo về mã:

• Mã là xác thực và thực sự có nguồn gốc bởi nhà xuất bản.

• Mã không được sửa đổi kể từ khi nó rời khỏi nhà xuất bản phần mềm.

• Không thể phủ nhận nhà xuất bản đã xuất bản mã. Điều này cung cấp sự không chối bỏ từ hành động xuất bản.

Tiêu chuẩn Xử lý Thông tin Liên bang của Chính phủ Hoa Kỳ (FIPS) Ấn bản 140-3, quy định rằng phần mềm có sẵn để tải xuống trên internet phải được ký điện tử và xác minh. Mục đích của phần mềm được ký điện tử là để đảm bảo rằng phần mềm đó không bị giả mạo và nó có nguồn gốc từ nguồn đáng tin cậy như đã tuyên bố. Chữ ký số đóng vai trò xác minh rằng mã không bị can thiệp bởi các tác nhân đe dọa và mã độc hại không bị bên thứ ba chèn vào tệp.

1. **Digital Signatures for Digital Certificates**

Chứng thư số tương đương với hộ chiếu điện tử. Nó cho phép người dùng, máy chủ và tổ chức trao đổi thông tin qua Internet một cách an toàn. Cụ thể, chứng chỉ kỹ thuật số được sử dụng để xác thực và xác minh rằng người dùng đang gửi tin nhắn là người mà họ tuyên bố. Chứng chỉ kỹ thuật số cũng có thể được sử dụng để cung cấp tính bảo mật cho người nhận bằng các phương tiện để mã hóa thư trả lời.

Chứng chỉ SSL là một chứng chỉ kỹ thuật số xác nhận danh tính của một miền trang web. Để triển khai SSL trên trang web của mình, bạn mua chứng chỉ SSL cho miền của mình từ nhà cung cấp Chứng chỉ SSL. Bên thứ ba đáng tin cậy thực hiện một cuộc điều tra chuyên sâu trước khi cấp thông tin xác thực. Sau cuộc điều tra chuyên sâu này, bên thứ ba sẽ cấp thông tin xác thực (tức là chứng chỉ số) rất khó giả mạo. Từ thời điểm đó trở đi, tất cả các cá nhân tin tưởng bên thứ ba chỉ cần chấp nhận thông tin xác thực mà bên thứ ba cấp. Khi máy tính cố gắng kết nối với một trang web qua HTTPS, trình duyệt web sẽ kiểm tra chứng chỉ bảo mật của trang web và xác minh rằng chứng chỉ đó hợp lệ và có nguồn gốc từ CA đáng tin cậy. Điều này xác nhận rằng trang web xác định là đúng. Chứng chỉ được lưu cục bộ bởi trình duyệt web và sau đó được sử dụng trong các giao dịch tiếp theo. Khóa công khai của trang web được bao gồm trong chứng chỉ và được sử dụng để xác minh các giao tiếp trong tương lai giữa trang web và khách hàng.

Các bên thứ ba đáng tin cậy này cung cấp các dịch vụ tương tự như các văn phòng cấp phép của chính phủ.

Cơ sở hạ tầng khóa công khai (PKI - Public Key Infrastructure) bao gồm các đặc điểm kỹ thuật, hệ thống và công cụ được sử dụng để tạo, quản lý, phân phối, sử dụng, lưu trữ và thu hồi chứng chỉ kỹ thuật số. Cơ quan cấp chứng chỉ (CA) là tổ chức tạo chứng chỉ kỹ thuật số bằng cách gắn khóa công khai vào danh tính đã xác nhận, chẳng hạn như trang web hoặc cá nhân. PKI là một hệ thống phức tạp được thiết kế để bảo vệ danh tính kỹ thuật số khỏi bị tấn công bởi ngay cả những tác nhân đe dọa tinh vi nhất hoặc các quốc gia.

Một số ví dụ về Tổ chức phát hành chứng chỉ là IdenTrust, DigiCert, Sectigo, GlobalSign và GoDaddy. Các CA này tính phí cho các dịch vụ của họ. Let’s Encrypt là một CA phi lợi nhuận cung cấp chứng chỉ miễn phí.

PKI là cần thiết để hỗ trợ phân phối quy mô lớn và xác định các khóa mã hóa công khai. Khuôn khổ PKI tạo điều kiện cho một mối quan hệ tin cậy có khả năng mở rộng cao.

Nó bao gồm phần cứng, phần mềm, con người, chính sách và thủ tục cần thiết để tạo, quản lý, lưu trữ, phân phối và thu hồi chứng chỉ kỹ thuật số.

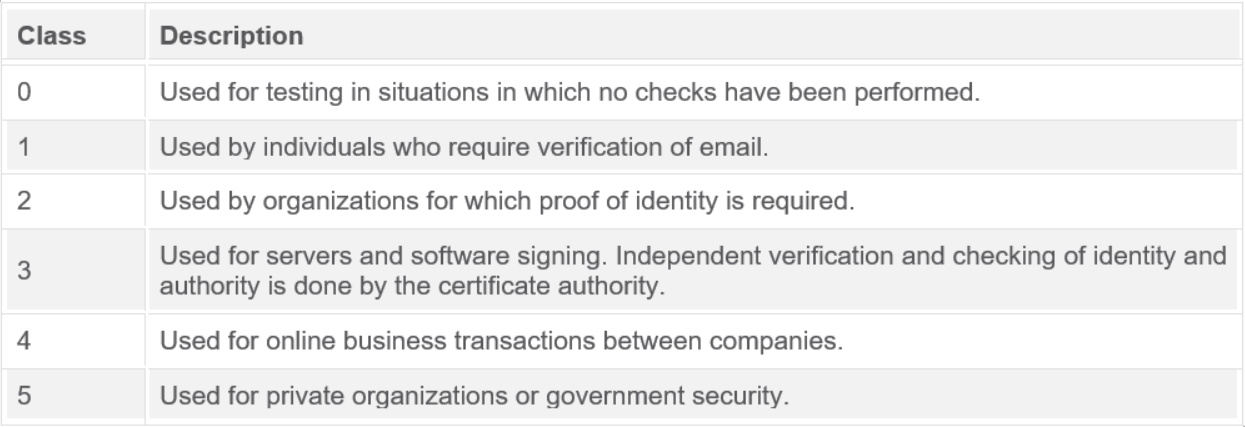
1. **The PKI Authorities System**

Nhiều nhà cung cấp cung cấp máy chủ CA như một dịch vụ được quản lý hoặc như một sản phẩm của người dùng cuối. Một số nhà cung cấp này bao gồm Symantec Group (VeriSign), Comodo, Go Daddy Group, GlobalSign và DigiCert trong số những nhà cung cấp khác.

Các tổ chức cũng có thể triển khai PKI riêng tư bằng Microsoft Server hoặc Open SSL.

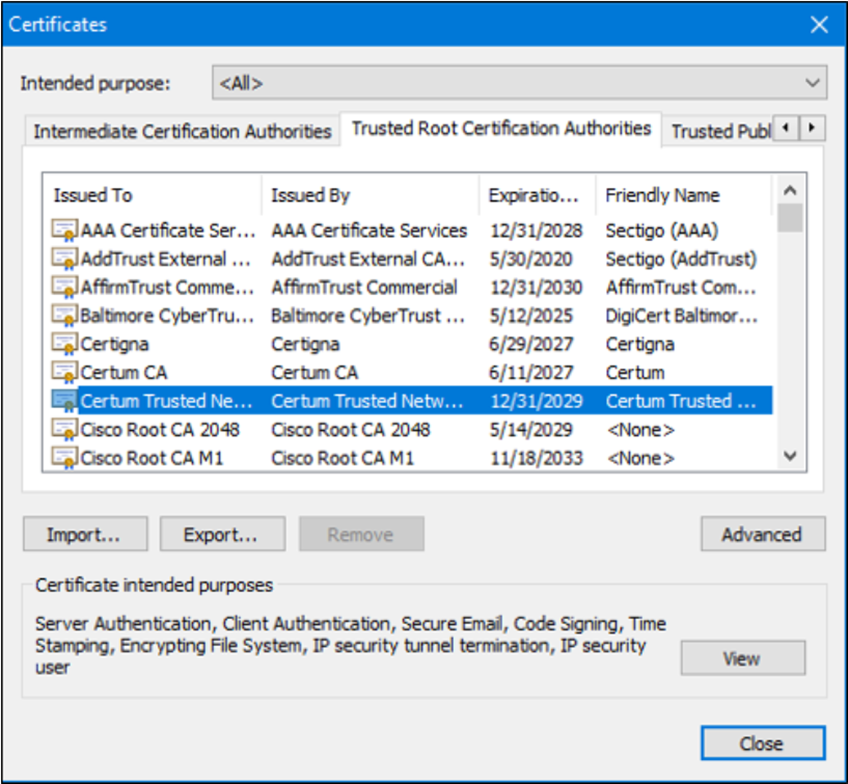
CA, đặc biệt là những CA được thuê ngoài, cấp chứng chỉ dựa trên các lớp xác định mức độ đáng tin cậy của chứng chỉ.

Bảng cung cấp mô tả về các lớp. Số lớp được xác định bởi mức độ nghiêm ngặt của quy trình đã xác minh danh tính của chủ sở hữu khi chứng chỉ được cấp. Số lớp càng cao, chứng chỉ càng được tin cậy. Do đó, chứng chỉ hạng 5 được tin tưởng hơn nhiều so với chứng chỉ hạng thấp hơn.



Ví dụ: chứng chỉ hạng 1 có thể yêu cầu trả lời email từ chủ sở hữu để xác nhận rằng họ muốn đăng ký. Loại xác nhận này là xác thực yếu của chủ sở hữu. Đối với chứng chỉ hạng 3 hoặc 4, người sở hữu trong tương lai phải chứng minh danh tính và xác thực khóa công khai bằng cách xuất trình trực tiếp với ít nhất hai tài liệu ID chính thức.

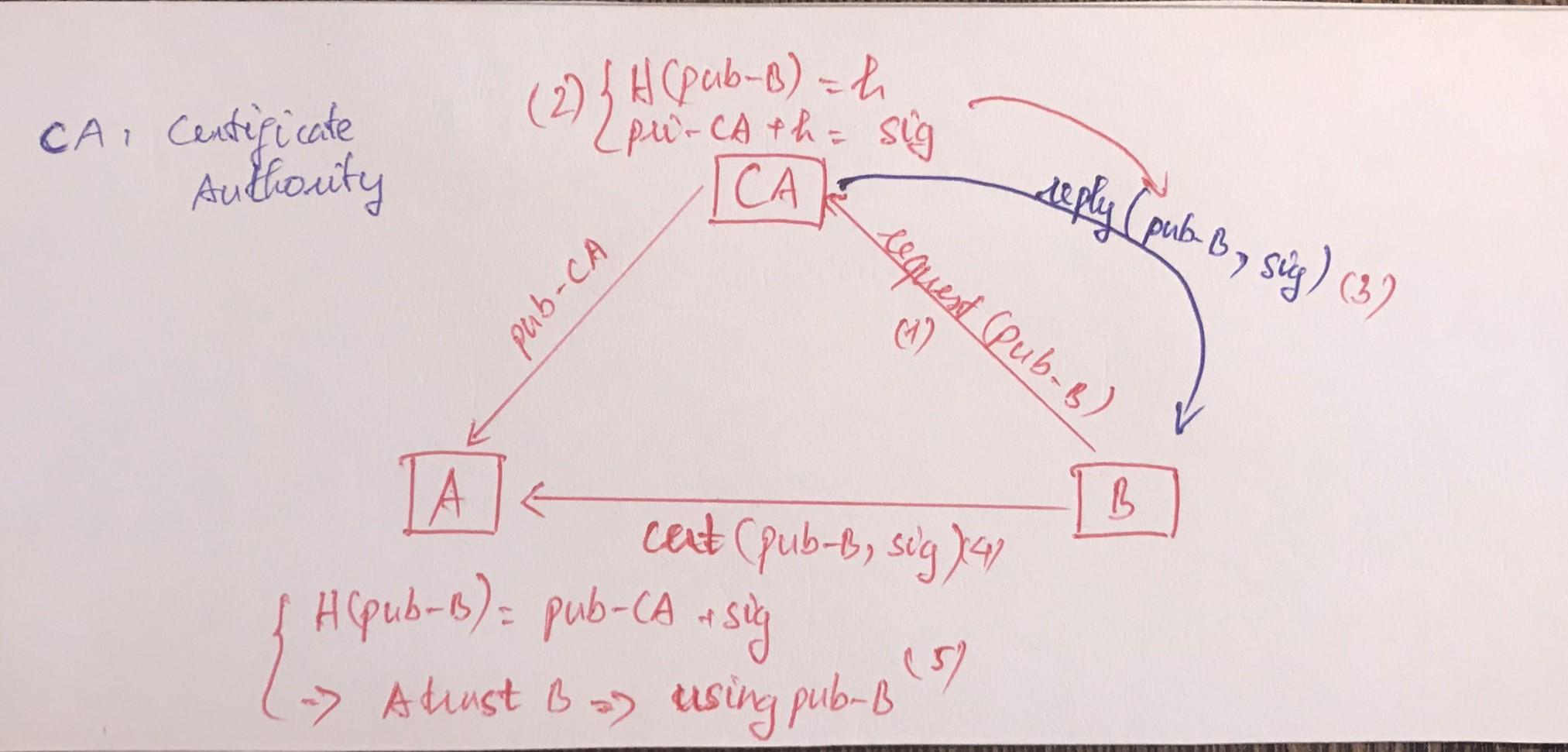
Một số khóa công khai CA được tải trước, chẳng hạn như những khóa được liệt kê trong trình duyệt web. Hình này hiển thị các chứng chỉ VeriSign khác nhau có trong kho chứng chỉ trên máy chủ. Mọi chứng chỉ được ký bởi bất kỳ CA nào trong danh sách sẽ được trình duyệt coi là hợp pháp và sẽ tự động được tin cậy.



**Note**: Doanh nghiệp cũng có thể triển khai PKI để sử dụng nội bộ. PKI có thể được sử dụng để xác thực nhân viên đang truy cập mạng. Trong trường hợp này, doanh nghiệp là CA của chính nó.

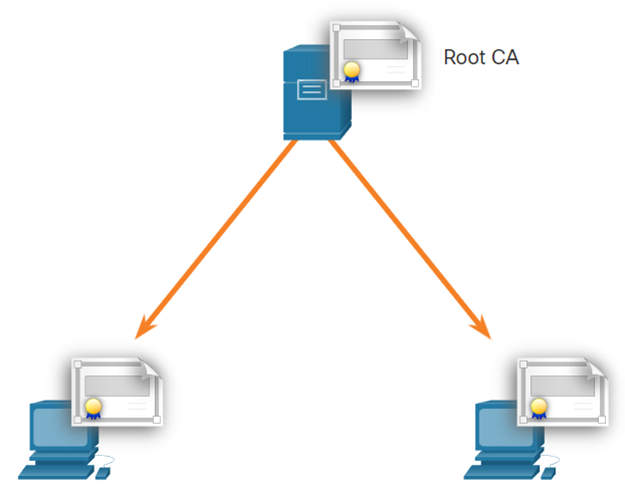
1. **The PKI Trust System**

Các bước để cung cấp chúng chỉ số cho các doanh nghiệp từ một CA đáng tin cậy thuộc kiểm soát của chính phủ. (step-by-step)



PKI có thể hình thành các cấu trúc liên kết tin cậy khác nhau. Đơn giản nhất là cấu trúc liên kết PKI gốc đơn.

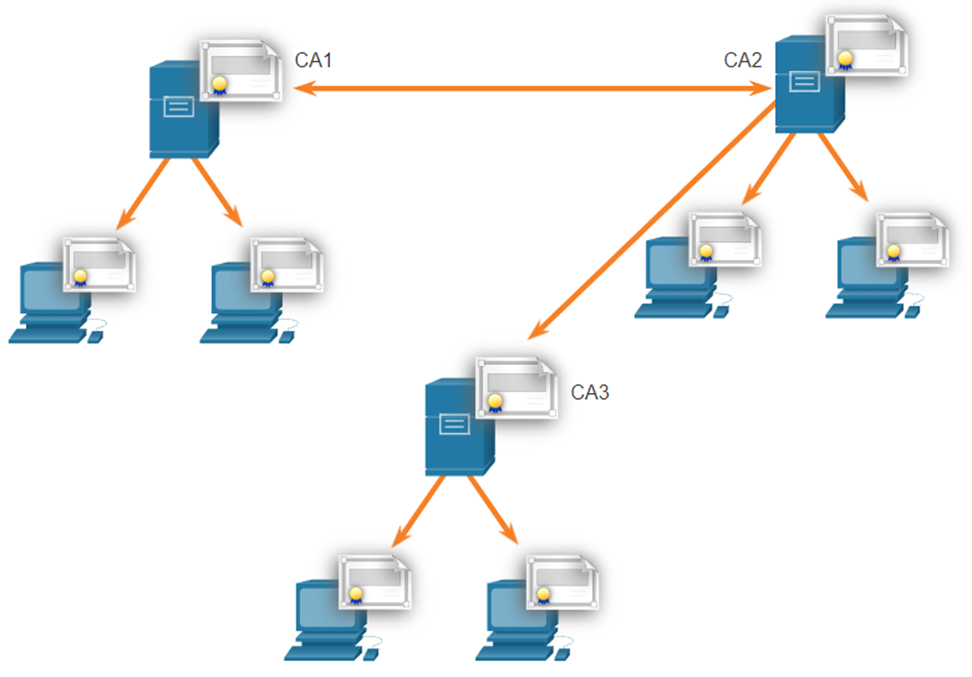
Như thể hiện trong hình bên dưới, một CA duy nhất, được gọi là CA gốc, cấp tất cả các chứng chỉ cho người dùng cuối, những chứng chỉ này thường nằm trong cùng một tổ chức. Lợi ích của cách tiếp cận này là tính đơn giản của nó. Tuy nhiên, rất khó để mở rộng quy mô ra một môi trường rộng lớn vì nó đòi hỏi sự quản lý tập trung chặt chẽ, điều này tạo ra một điểm thất bại duy nhất.



Trên các mạng lớn hơn, PKI CA có thể được liên kết bằng hai kiến trúc cơ bản:

Cấu trúc liên kết CA được chứng nhận chéo - Như thể hiện trong hình bên dưới, đây là mô hình ngang hàng trong đó các CA riêng lẻ thiết lập mối quan hệ tin cậy với các CA khác bằng cách chứng nhận chéo các chứng chỉ CA. Người dùng trong cả hai miền CA cũng được đảm bảo rằng họ có thể tin tưởng lẫn nhau. Điều này cung cấp khả năng dự phòng và loại bỏ điểm lỗi duy nhất.

Hình bên cho thấy thiết lập tương tự như cấu trúc liên kết PKI đơn căn trước đó, nhưng nó được gắn nhãn CA1. có một mũi tên hai chiều giữa cấu trúc liên kết này và một cấu trúc liên kết khác có nhãn CA2. một mũi tên chỉ từ cấu trúc liên kết CA2 đến một cấu trúc liên kết khác có nhãn CA3.



Cấu trúc liên kết CA phân cấp - Như trong hình bên dưới, CA cấp cao nhất được gọi là CA gốc. Nó có thể cấp chứng chỉ cho người dùng cuối và CA cấp dưới. Các CA phụ có thể được tạo để hỗ trợ các đơn vị kinh doanh, miền hoặc cộng đồng tin cậy khác nhau. CA gốc duy trì “cộng đồng tin cậy” đã được thiết lập bằng cách đảm bảo rằng mỗi thực thể trong hệ thống phân cấp đều tuân thủ một tập hợp các thông lệ tối thiểu. Các lợi ích của cấu trúc liên kết này bao gồm tăng khả năng mở rộng và khả năng quản lý. Cấu trúc liên kết này hoạt động tốt trong hầu hết các tổ chức lớn. Tuy nhiên, có thể khó xác định chuỗi của quá trình ký kết.

Cấu trúc liên kết phân cấp và chứng nhận chéo có thể được kết hợp để tạo ra một cơ sở hạ tầng kết hợp. Một ví dụ sẽ là khi hai cộng đồng phân cấp muốn chứng nhận chéo lẫn nhau để các thành viên của mỗi cộng đồng tin tưởng lẫn nhau.

Hình bên cho thấy một máy chủ có nhãn CA root với một chứng chỉ bên cạnh nó. Có hai mũi tên chỉ đến một CA cấp dưới, mỗi mũi tên có một cấu trúc liên kết PKI gốc đơn.

