TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI TP. HỒ CHÍ MINH

KHOA CÔNG NGHỆ THÔNG TIN

AN TOÀN THÔNG TIN- INFORMATION SECURITY

CHƯƠNG 3
KỸ THUẬT MÃ HOÁ
BÀI 8
MÃ HOÁ HIỆN ĐẠI – MÃ HÓA BẤT ĐỐI XỨNG

Giảng viên: TS. Trần Thế Vinh

MÃ HOÁ HIỆN ĐẠI

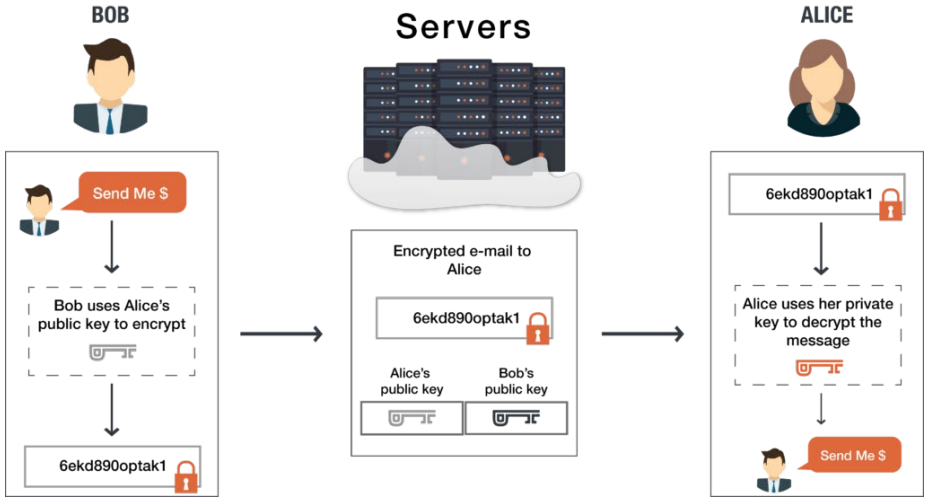
MÃ HOÁ KHOÁ BẤT ĐỐI XỨNG

Khái niệm:

Mã hóa khóa bất đối xứng (thường được gọi là mã hóa khóa công khai) sử dụng một cặp khóa cho quá trình mã hóa và giải mã. Mặc dù các khóa khác nhau nhưng chúng có liên quan về mặt toán học, do đó việc truy xuất bản rõ bằng cách giải mã bản mã là khả thi.

Đặc điểm nổi bật của các hệ mã khóa bất đối xứng là kích thước khóa rất lớn, có thể lên đến hàng ngàn bit. Do vậy, các hệ mã hóa dạng này thường có tốc độ thực thi chậm hơn nhiều lần so với các hệ mã hóa khóa đối xứng có độ an toàn tương đương.

Mặc dù vậy, các hệ mã hóa khóa bất đối xứng có khả năng đạt độ an toàn cao và ưu điểm nổi bật nhất là việc quản lý và phân phối khóa đơn giản hơn do chỉ khóa riêng trong cặp khóa cần giữ bí mật, còn khóa công khai có thể phân phối rộng rãi trong môi trường mở như mạng Internet.



MÃ HOÁ HIỆN ĐẠI

MÃ HOÁ KHOÁ BẤT ĐỐI XỨNG – CƠ SỞ TOÁN HỌC

Ba loại chính của thuật toán mã hóa bất đối xứng:

1. Phân tích số nguyên:

Các thuật toán phân tích nhân tử số nguyên dựa trên thực tế là số nguyên lớn rất khó phân tích thành nhân tử. Một ví dụ điển hình của thuật toán như vậy là RSA.

2. Logarit rời rạc:

Thuật toán dựa trên logarit rời rạc là bài toán số học modulo. Tính phần dư của một phép chia rất dễ dàng. Đồng thời, việc tìm kiếm số mũ của căn nguyên là một công việc rất tốn công sức. Nói cách khác, việc tìm dữ liệu đầu vào mà chỉ biết kết quả là cực kỳ khó khăn. Đây là chức năng một chiều. Hãy xem xét phương trình sau:

$$3^4 \equiv 13 \ (mod\ 17)$$

Khi biết kết quả của phương trình, việc tính số mũ của 3 hay giải phương trình $3^k \equiv 13 \ (mod\ 17)$ (không có 1 cơ sở nào để tính số mũ theo cách chung, chỉ sử dụng phép nhân thử). Sự phức tạp của phép tính như vậy làm cơ sở cho thuật toán Diffie-Hellman và các thuật toán chữ ký số

3. Đường cong Elliptic:

Thuật toán đường cong Elliptic xem xét bài toán logarit rời rạc được mô tả ở trên trong ngữ cảnh của đường cong elliptic. Đường cong elliptic là một đường cong đại số trên một trường. Nó là một đường cong không kỳ dị: nó không có đỉnh hoặc các điểm tự giao nhau. Đường cong có một điểm ở vô cùng và 2 tham số (a, b).
Trong trường hợp này, a và b là các số nguyên có giá trị được bao gồm trong trường mà đường cong elliptic được vẽ. Đường cong Elliptic có thể được xây dựng trên các số thực, số hữu tỉ và số phức, cũng như trên các trường hữu hạn. Trong mật mã, thay vì các số thực, các đường cong Elliptic được sử dụng xây dựng trên các trường hữu hạn đơn giản. Trong trường hợp này, đặc tính trường phải lớn hơn 3. Bằng cách thay đổi các tham số a và/hoặc b, ta có thể xây dựng các đường cong Elliptic khác nhau

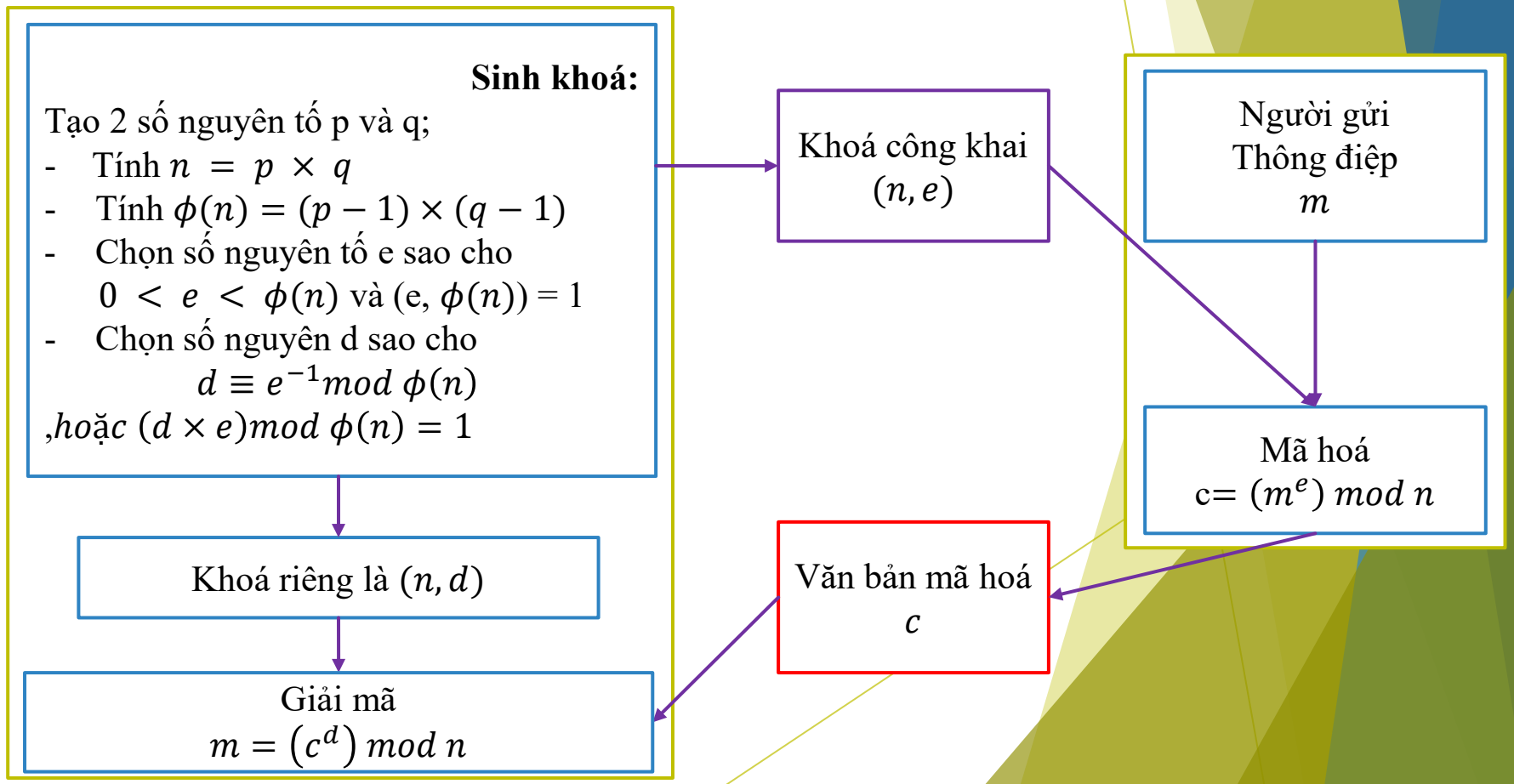
MÃ HOÁ HIỆN ĐẠI

MÃ HOÁ KHOÁ BẤT ĐỐI XỨNG – THUẬT TOÁN RSA

Thuật toán RSA:

Đây có lẽ là thuật toán mã hóa bất đối xứng nổi tiếng và được sử dụng rộng rãi nhất. Trên thực tế, chính thuật toán này đóng vai trò là nền tảng cho các công cụ mật mã sinh học, trong đó các nguyên tắc mật mã có thể được sử dụng để bảo vệ mẫu sinh trắc học hơn nữa. Thuật toán RSA bắt nguồn từ tập đoàn dữ liệu RSA và nó được đặt tên theo những người phát minh ra nó (Ron Rivest, Ali Shamir, Leonard Adelman).

Thuật toán RSA sử dụng sức mạnh của các số nguyên tố để tạo cả khóa chung và khóa riêng. Các quy trình tạo khóa, mã hóa và giải mã cho thuật toán được trình bày như hình bên.



MÃ HOÁ HIỆN ĐẠI

MÃ HOÁ KHOÁ BẤT ĐỐI XỨNG – THUẬT TOÁN RSA

Yêu cầu với các tham số sinh khóa p và q như sau:

- Các số nguyên tố p và q phải được chọn sao cho việc phân tích n (với $n = p.q$) là không khả thi về mặt tính toán;
- p và q nên có cùng độ lớn (tính bằng bit) và phải là các số đủ lớn. Nếu n có kích thước 2048 bit thì p và q nên có kích thước khoản 1024 bit.
- Hiệu số $p - q$ không nên quá nhỏ, do nếu $p - q$ quả nhỏ, tức là $p \approx q$ và $p \approx \sqrt{n}$. Như vậy, có thể chọn các số nguyên tố ở gần $\sqrt{n}$ và thử lần lượt.
- Khi có được p, có thể tính q và tìm ra d là khóa bí mật từ khóa công khai e và
  . Nếu p và q được chọn ngẫu nhiên và $p - q$ đủ lớn, khả năng hai số này bị phân tích từ n sẽ giảm đi.

Các số nguyên tố p và q nên là số nguyên tố mạnh (strong prime). Số nguyên tố p được xem là một số nguyên tố mạnh nếu nó thỏa mãn 3 điều kiện sau:

- ✓ $p - 1$ có một thừa số nguyên tố lớn, giả thiết là r;
- ✓ $p + 1$ có một thừa số nguyên tố hớn;
- ✓ $r - 1$ cũng có một thừa số nguyên tố lớn;

MÃ HOÁ HIỆN ĐẠI

MÃ HOÁ KHOÁ BẤT ĐỐI XỨNG – THUẬT TOÁN RSA

Ví dụ:

Sinh khóa:

Chọn 2 số nguyên tố p = 53, q = 73
Tính n=p.q=53.73=3869
Tính $\phi(n) = (p-1).(q-1) = 52.72 = 3744.$
Chọn số e sao cho 0<e<3744, và $\gcd(e, \phi(n)) = 1$. Chọn e = 5
Tính $(d.e) mod \phi(n) = 1 \Longrightarrow (d.5) mod 3744 = 1 \Longrightarrow d = 749$ (với k=1)

Do đó ta có: Khóa công khai là (3869, 5), và khóa riêng là (3869, 749)

Mã hóa:

Với bản rõ "HI" tức là: "H"=8, "I"=9, có nghĩa là m = 89
$c = m^e mod\ n = 89^5 mod\ 3869 = 1391$
Vậy bản mã c=1391

Giải mã:

Với bản mã c=1391
$m = c^d mod\ n = 1391^{749} mod\ 3869 = 89$
Vậy bản rõ m=89 tương ứng với "HI"

MÃ HOÁ HIỆN ĐẠI

MÃ HOÁ KHOÁ BẤT ĐỐI XỨNG – THUẬT TOÁN DIFFIE - HELLMAN

Thuật toán Diffie-Hellman:

Về thuật toán bất đối xứng Diffie Hellman, nó cũng được đặt theo tên của những người phát minh ra nó (White Diffie và Martin Hellman. Nó còn được gọi là "Thuật toán DH". Tuy nhiên, thuật toán này không được sử dụng để mã hóa bản mã mà thay vào đó, mục tiêu chính của nó là tìm ra giải pháp để gửi tổ hợp khóa chung/khóa riêng thông qua 1 kênh an toàn.

Cách hoạt động của thuật toán Diffie-Hellman như sau:

- Bên nhận có quyền sở hữu khóa chung và khóa riêng đã được tạo, nhưng lần này chúng được tạo bởi thuật toán DH.
- Bên gửi(sender) nhận khóa chung do bên nhận(receiver) tạo và do đó sử dụng thuật toán DH để tạo một bộ khóa chung khác, nhưng trên cơ sở tạm thời.
- Bên gửi (sender) hiện sử dụng tổ hợp khóa công khai/khóa riêng tạm thời mới được tạo này, do bên nhận (receiver) gửi để tạo một số bí mật, ngẫu nhiên – số này được gọi cụ thể là "khóa phiên(key session)"
- Bên gửi (sender) sử dụng khóa phiên mới được thiết lập này để mã hóa thêm thông điệp bản mã và gửi chuyển tiếp tới bên nhận (receiver), với khóa chung được tạo tạm thời.
- Khi bên nhận cuối cùng cũng nhận được thông điệp bản mã từ bên gửi (sender), khóa phiên có thể được suy ra bằng toán học.
- Sau khi hoàn thành bước trên, bên nhận (receiver) có thể giải mã phần còn lại của bản mã.

MÃ HOÁ HIỆN ĐẠI

MÃ HOÁ KHOÁ BẤT ĐỐI XỨNG – THUẬT TOÁN DIFFIE - HELLMAN

Diễn giải thuật toán Diffie-Hellman:

Alice	Bob
Khóa công khai có sẵn P,G	Khóa công khai có sẵn P,G
Khóa riêng được chọn a	Khóa riêng được chọn b
Khóa được tạo ra $x = G^a mod\ P$	Khóa được tạo ra $y = G^b mod\ P$
Trao đổi các khóa được tạo diễn ra giữa Alice và Bob	
Khóa đã nhận y	Khóa đã nhận x
Khóa bí mật được tạo $k_a = y^a mod\ P$	Khóa bí mật được tạo $k_b = x^b mod\ P$
Về mặt đại số, ta có thể chỉ ra rằng $k_a = k_b$	

MÃ HOÁ HIỆN ĐẠI

MÃ HOÁ KHOÁ BẤT ĐỐI XỨNG – THUẬT TOÁN DIFFIE - HELLMAN

Ví dụ:

Bước 1: Alice và Bob lấy số công khai P = 23, G = 9

Bước 2: Alice chọn khóa riêng a = 4 và
Bob chọn khóa riêng b = 3

Bước 3: Alice và Bob tính giá trị công khai
Alice: $x = (9^4 mod\ 23) = (6561\ mod\ 23) = 6$
Bob: $y = (9^3 mod\ 23) = (729\ mod\ 23) = 16$

Bước 4: Alice và Bob trao đổi số công khai Bước 5

Bước 5: Alice nhận khóa công khai y =16 và Bob nhận khóa công khai x = 6

Bước 6: Alice và Bob tính toán khóa đối xứng
Alice: $k_a = y^a mod\ p = 65536\ mod\ 23 = 9$
Bob: $k_b = x^b\ mod\ p = 216\ mod\ 23 = 9$

Bước 7: 9 là khóa bí mật được chia sẻ.

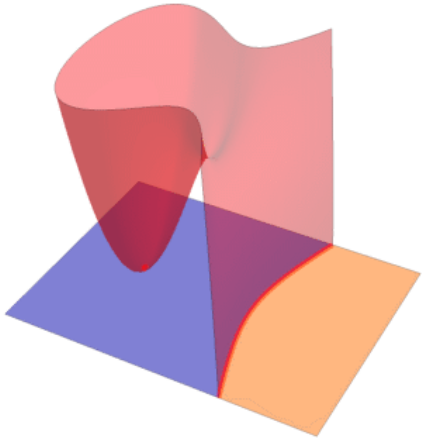
MÃ HOÁ HIỆN ĐẠI

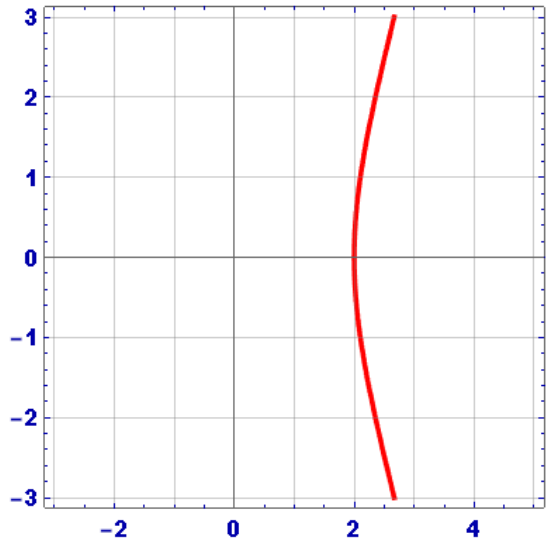
HỆ MẬT MÃ TRÊN ĐƯỜNG CONG ELLIPTIC (ECC)

Năm 1985, thuật toán mã hóa bất đối xứng mới được đề xuất dựa trên đường cong Elliptic. Mật mã đường cong Elliptic(ECC) dựa trên cấu trúc đại số của các đường cong elliptic trên các trường hữu hạn và độ khó của Bài toán logarit rời rạc trên đường cong Elliptic (ECDLP).

ECC thực hiện tất cả các khả năng chính của hệ thống mật mã bất đối xứng: mã hóa, chữ ký và trao đổi khóa.

Mật mã ECC được coi là sự kế thừa tự nhiên hiện đại của hệ thống mật mã RSA, bởi vì ECC sử dụng các khóa và chữ ký nhỏ hơn RSA cho cùng 1 mức độ bảo mật và cung cấp khả năng tạo khóa rất nhanh, thỏa thuận khóa nhanh và chữ ký nhanh.





MÃ HOÁ HIỆN ĐẠI

HỆ MẬT MÃ TRÊN ĐƯỜNG CONG ELLIPTIC (ECC)

Khóa ECC:

Các khóa riêng trong ECC là số nguyên (trong phạm vi kích thước trường của đường cong, thường là số nguyên 256 bit).

Ví dụ về khóa riêng ECC 256 bit (được mã hóa hex, 32 byte hay 64 chữ số hex):

0x51897b64e85c3f714bba707e867914295a1377a7463a9dae8ea6a8b914246319

Việc tạo khóa trong mật mã ECC cũng đơn giản như tạo một số nguyên ngẫu nhiên một cách an toàn trong phạm vi nhất định, vì vậy quá trình này cực kỳ nhanh. Bất kỳ số nào trong phạm vi đều là khóa riêng ECC hợp lệ.

Các khóa công khai trong ECC là các điểm EC – cặp tọa độ nguyên {x, y}, nằm trên đường cong. Do tính chất đặc biệt của chúng, các điểm EC có thể được nén thành một toạ độ + 1 bit (lẽ hoặc chẵn). Do đó, khóa chung được nén, tương ứng với khóa riêng ECC 256 bit, là một số nguyên 257 bit (256 +1). Ví dụ về khóa công khai ECC (tương ứng với khóa riêng ở trên được mã hóa ở định dạng Ethereum, dưới dạng hex với tiền tố 02 hoặc 03) là:
0x02f54ba86dc1ccb5bed0224d23f01ed87e4a443c47fc690d7797a13d41d2340e1a
Ở định dạng này khóa chung chiếm 33 byte (66 chữ số hex), có thể được tối hưu hóa thành chính xác 257 bit.

MÃ HOÁ HIỆN ĐẠI

HỆ MẬT MÃ TRÊN ĐƯỜNG CONG ELLIPTIC (ECC)

Đường cong và độ dài khóa:

Các thuật toán mã hóa ECC có thể sử dụng các đường cong elliptic cơ bản khác nhau. Các đường cong khác nhau cung cấp mức bảo mật khác nhau (độ mạnh của mật mã), hiệu suất khác nhau (tốc độ mã hóa/giải mã) và độ dài khóa khác nhau. Cũng có thể liên quan đến các thuật toán khác nhau.

Đường cong ECC được sử dụng trong các tiêu chuẩn bảo mật và thư viện mật mã phổ biến có tên gọi là (secp256k1 hoặc Curve25519), kích thước trường (xác định độ dài khóa, 256 bit), cường độ bảo mật(kích thước trường /2 hoặc bé hơn), hiệu suất (phép tính/s) và một số tham số khác.

Các khóa ECC có độ dài phụ thuộc trực tiếp vào đường cong elliptic. Trong hầu hết các ứng dụng (như OpenSSL, OpenSSH, Bitcoin), độ dài mặc định của khóa riêng ECC là 256 bit, nhưng tùy thuộc vào đường cong, có thể có nhiều kích thước khóa ECC khác nhau: 192 bit (đường cong secp192r1), 233 bit (đường cong sect233k1), 224 bit (đường cong secp224k1), 256 bit (đường cong secp256k1, curve25519), 283 bit (đường cong sect283k1), 384 bit (đường cong p384 và secp384r1). 409 bit (đường cong sect409r1), 414 bit (đường cong Curve41417), 448 bit (đường cong Curve448-Goldilocks), 511 bit (đường cong M-511), 521 bit (đường cong P-521), 571 bit (đường cong sect571k1) và nhiều loại khác.

MÃ HOÁ HIỆN ĐẠI

HỆ MẬT MÃ TRÊN ĐƯỜNG CONG ELLIPTIC (ECC)

Thuật toán ECC:

Mật mã đường cong elliptic (ECC) cung cấp một nhóm thuật toán dựa trên toán học của các đường cong elliptic trên các trường hữu hạn:

- Các thuật toán chữ ký số ECC như ECDSA (đối với đường cong cổ điển) và EdDSA (đối với đường cong Edwards xoắn).
- Các thuật toán mã hóa ECC và sơ đồ mã hóa kết hợp như sơ đồ mã hóa tích hợp ECIES và EEECC (ElGamal dựa trên EC)
- Các thuật toán thỏa thuận khóa ECC như ECDH, X25519, FHMQV.

Tất cả các thuật toán này sử dụng một đường cong phía sau (như secp256k1, Curve25519 hoặc p521) để tính toán và dựa vào độ khó của ECDLP (bài toán logarit rời rạc đường cong elliptic). Tất cả các thuật toán này đều sử dụng cặp khóa chung/riêng, trong đó khóa riêng là một số nguyên và khóa chung là một điểm trên đường cong elliptic (điểm EC).

MÃ HOÁ HIỆN ĐẠI

HỆ MẬT MÃ TRÊN ĐƯỜNG CONG ELLIPTIC (ECC)

Đường cong Elliptic:

Trong toán học, đường cong elliptic là đường cong đại số phẳng, bao gồm tất cả các điểm {x, y}, được biểu diễn bởi phương trình:

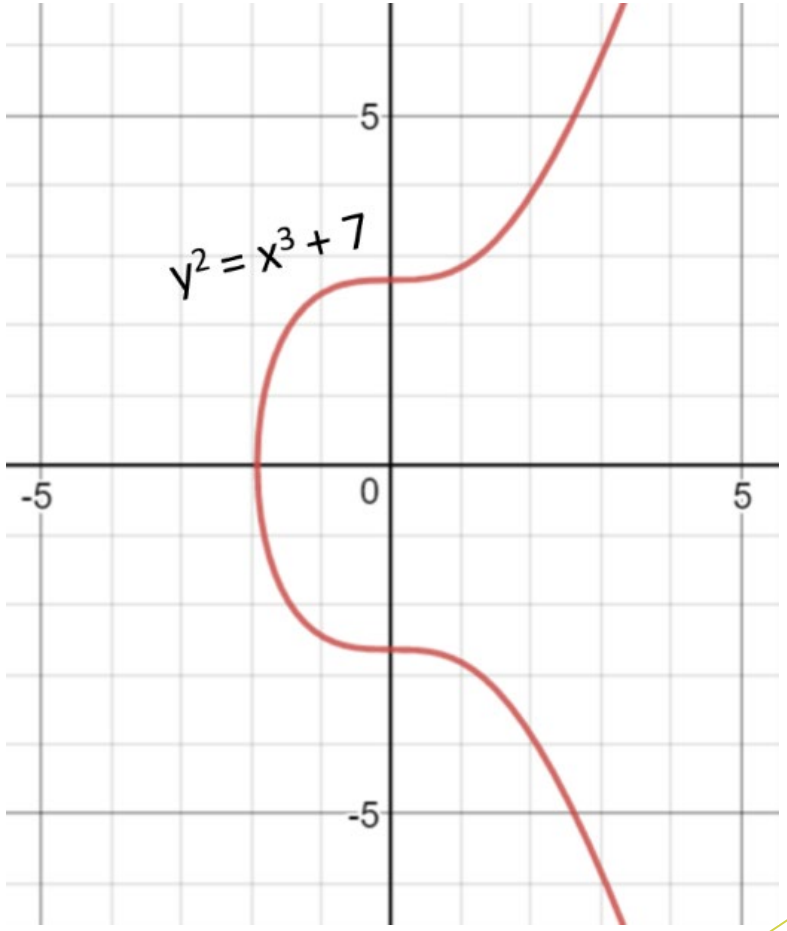
$$Ax^3 + Bx^2y + Cxy^2 + Dy^3 + Ex^2 + Fxy + Gy^2 + Hx + Iy + J = 0$$

Mật mã học sử dụng các đường cong elliptic ở dạng đơn giản hóa (dạng Weierstras), được định nghĩa bằng phương trình sau:

$$y^2 = x^3 + ax + b \quad (1)$$

Ví dụ: đường cong NIST secp256k1 (được sử dụng trong Bitcoin) dựa trên đường cong elliptic có dạng:

$y^2 = x^3 + 7$ (phương trình đường cong elliptic (1), trong đó, a=0 và b=7)



MÃ HOÁ HIỆN ĐẠI

HỆ MẬT MÃ TRÊN ĐƯỜNG CONG ELLIPTIC (ECC)

Đường cong Elliptic trên trường hữu hạn:

Mật mã đường cong elliptic (ECC) sử dụng các đường cong elliptic trên trường hữu hạn G_p (trong đó p là số nguyên tố và p>3) hoặc G_{2^m} (trong đó kích thước trường $p = 2^m$). Điều này có nghĩa là trường là một ma trận vuông có kích thước pxp và các điểm trên đường cong chỉ được giới hạn ở các toạ độ nguyên trong trường. Tất cả các phép toán đại số trong trường (như cộng và nhân) đều dẫn đến một điểm khác trong trường. Phương trình đường cong elliptic trên trường hữu hạn G_p có dạng modulo như sau:

$$y^2 \equiv x^3 + ax + b(mod\ p)$$

Tương ứng, "đường cong BTC" secp256k1 có dạng:

$$y^2 \equiv x^3 + 7(mod\ p)$$

Không giống như RSA, sử dụng các số nguyên trong khoảng {0, ..., p-1} (trường Z_p) làm không gian khóa của nó. ECC sử dụng các điểm {x, y} trong trường Galois G_p (trong đó x và y là số nguyên trong khoảng {0,...,p-1}

Một đường cong elliptic trên trường hữu hạn G_p bao gồm:

- 1 tập hợp các tọa độ nguyên {x, y} sao cho $0 \leq x, y < p$
- Nằm trên đường cong elliptic $y^2 \equiv x^3 + ax + b(mod\ p)$

MÃ HOÁ HIỆN ĐẠI

HỆ MẬT MÃ TRÊN ĐƯỜNG CONG ELLIPTIC (ECC)

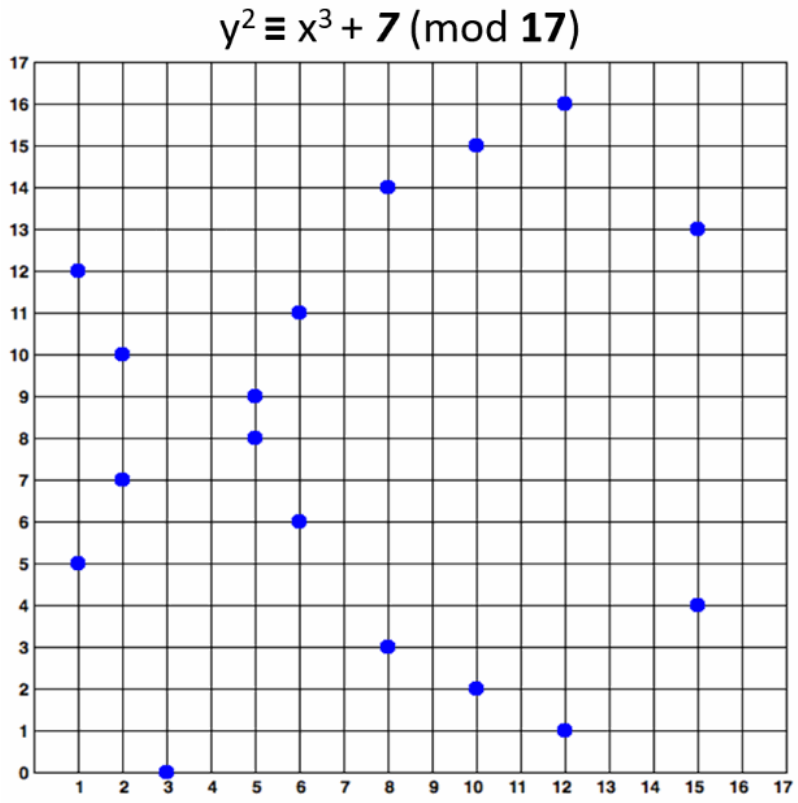
Đường cong Elliptic trên trường hữu hạn:

Ví dụ về đường cong elliptic trên trường hữu hạn F_{17}:

$$y^2 \equiv x^3 + 7(mod\ 17)$$

Lưu ý: đường cong elliptic trên trường hữu hạn $y^2 \equiv x^3 + 7(mod\ 17)$ bao gồm các điểm màu xanh ở hình bên, nghĩa là trong thực tế, "đường cong elliptic" được sử dụng trong mật mã là "tập hợp các điểm trong ma trận vuông", không phải là những "đường cong".

Đường cong trên cung cấp độ dài khóa rất nhỏ (4-5 bit). Trong thực tế, mật mã đường cong elliptic thường sử dụng các đường cong từ 256 bit trở lên.



MÃ HOÁ HIỆN ĐẠI

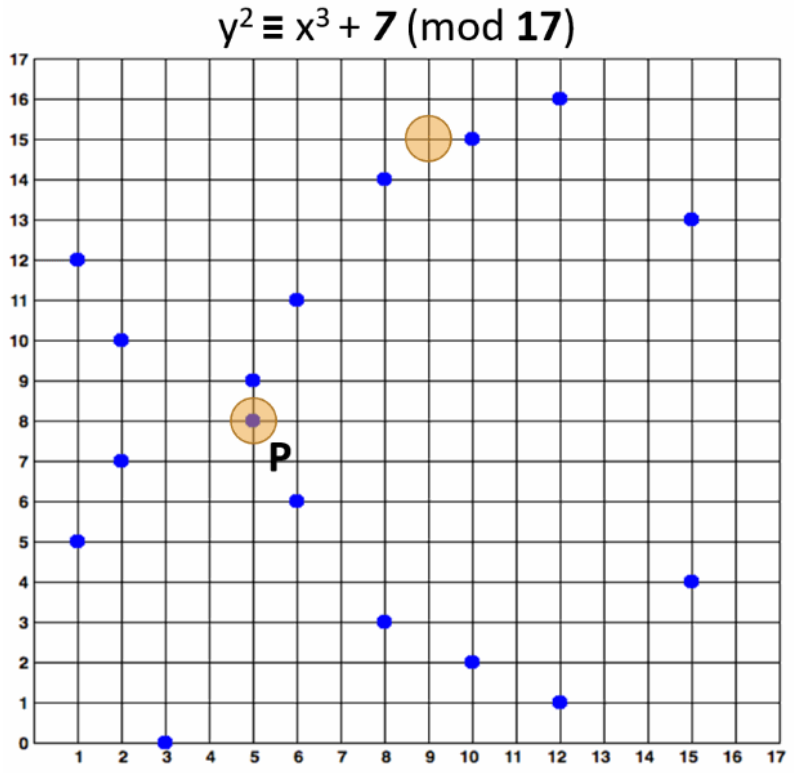
HỆ MẬT MÃ TRÊN ĐƯỜNG CONG ELLIPTIC (ECC)

Phép tính đường cong Elliptic trên trường hữu hạn:

Khá dễ dàng để tính xem một điểm nào đó có thuộc một đường cong elliptic nào đó trên trường hữu hạn hay không. Chẳng hạn, một điểm {x, y} thuộc đường cong $y^2 \equiv x^3 + 7 (mod\ 17)$ khi và chỉ khi:

$$x^3 + 7 - y^2 \equiv 0\ (mod\ 17)$$

Điểm P(5, 8) thuộc đường cong vì $(5^3 + 7 - 8^2) \equiv 0\ (mod\ 17)$. Điểm (9, 15) không thuộc đường cong.



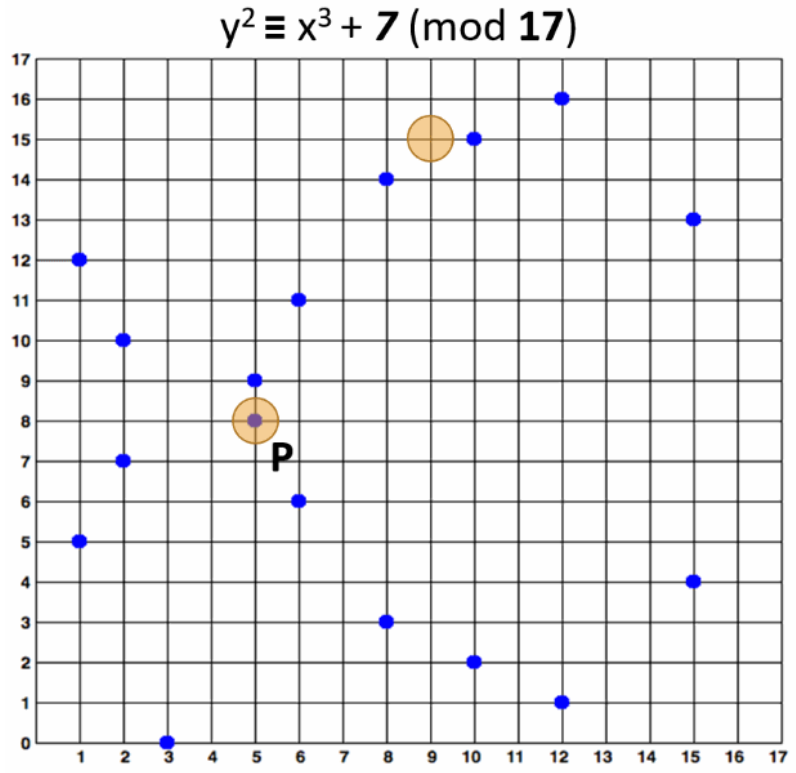
MÃ HOÁ HIỆN ĐẠI

HỆ MẬT MÃ TRÊN ĐƯỜNG CONG ELLIPTIC (ECC)

Nhân điểm ECC với số nguyên:

Hai điểm trên 1 đường cong elliptic (điểm EC) có thể được thêm vào và kết quả là một điểm khác. Thao tác này được gọi là cộng điểm EC. Nếu chúng ta thêm 1 điểm G vào chính nó, kết quả G+G=2G. Nếu chúng ta thêm G 1 lần nữa vào kết quả thì sẽ thu được 3G,..Đây là phép nhân điểm EC được xác định.

Một điểm G trên đường cong elliptic trên trường hữu hạn (điểm EC) có thể được nhân với một số nguyên k và kết quả là một điểm EC khác là P trên cùng 1 đường cong và thao tác này là rất nhanh:
P=kG



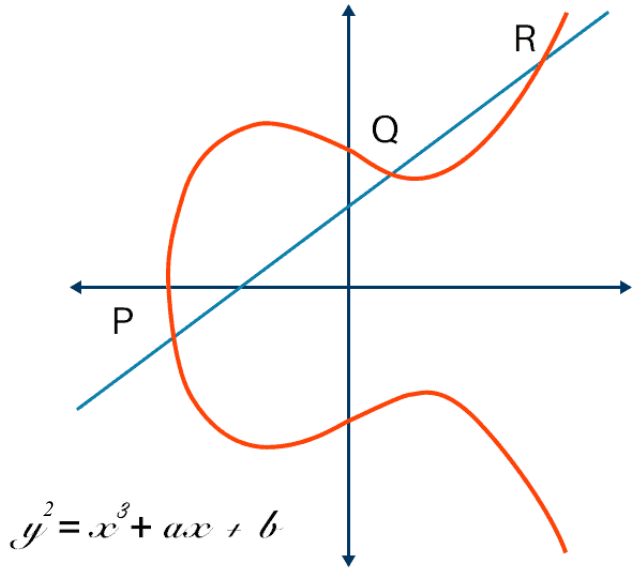
MÃ HOÁ HIỆN ĐẠI

HỆ MẬT MÃ TRÊN ĐƯỜNG CONG ELLIPTIC (ECC)

Các bước thực hiện mã hóa trên hệ đường cong elliptic như sau:

$$y^2 = x^3 + ax + b$$

1. Trao đổi khóa: -Global Public elements
   - $E_q(a, b)$ – đường cong elliptic với tham số a, b, q (là số nguyên tố hoặc số nguyên có dạng 2^m)
   - G – điểm tọa độ trong đường cong elliptic
     - ✓ Alice sinh khóa như sau:
       Chọn khóa riêng n_a: $n_a < n$
       Tính khóa công khai P_a: $P_a = n_a.G$
     - ✓ Bob sinh khóa như sau:
       Chọn khóa riêng n_b: $n_b < n$
       Tính khóa công khai P_b: $P_b = n_b.G$
     - ✓ Alice tính khóa bí mật: $k = n_a.P_b$
     - ✓ Bob tính khóa bí mật: $k = n_b.P_a$



2. Mã hóa:
   - Alice muốn chuyển cho Bob một tin nhắn M
   - Đầu tiên mã hóa tin nhắn M thành 1 điểm trên đường cong elliptic
   - Điểm này là P_m
   - Chọn số nguyên dương k bất kỳ
   - Khi đó điểm mật mã sẽ là:
     $$C_m = (k.G, P_m + kP_b)$$
   - Điểm này sẽ gửi đến Bob

3. Giải mã:
   - Để giải mã, ta nhân tọa độ x với khóa bí mật của người nhận (Bob)
     $$k.G.n_b$$
   - Sau đó trừ cho $(k.G.n_b)$ từ tọa độ y của điểm mật mã:
     $$P_m + k.P_b - (k.G.n_b)$$
   - Ta biết $P_b = n_b.G$ do đó:
     $$P_m + k.P_b - (k.G.n_b) = P_m + k.P_b - k.P_b = P_m$$
   - Bob nhận được điểm tin nhắn và có tin nhắn của Alice

MÃ HOÁ HIỆN ĐẠI

HỆ MẬT MÃ TRÊN ĐƯỜNG CONG ELLIPTIC (ECC)

Ví dụ:
Elliptic Curve Cryptography (ECC):

$$y^2 = x^3 + ax + b \quad (1)$$

Chọn $E_{13}(2,7)$, có nghĩa là a=2; b=7, q=13, $E_{13} = \{0, 1, 2, \dots, 12\}$
Lúc đó phương trình (1) có dạng: $y^2 = x^3 + 2x + 7(mod\ 13)$

x	$x^3 + 2x + 7(mod13)$		y^2	$y^2(mod13)$
0	7		0	0
1	10		1	1
2	6		2	4
3	1		3	9
4	1		4	3
5	12		5	12
6	1		6	10
7	0		7	10
8	2		8	12
9	0		9	3
10	0		10	9
11	8		11	4
12	4		12	1

Từ bảng bên ta thu được các điểm trên đường cong $y^2 = x^3 + 2x + 7$:

$(1,6), (1,7), (3,1), (3,12)$
$(4,1), (4,12), (5,5), (5,8)$
$(6,1), (6,12), (7,0), (9,0)$
$(10,0), (12,2), (12,11)$

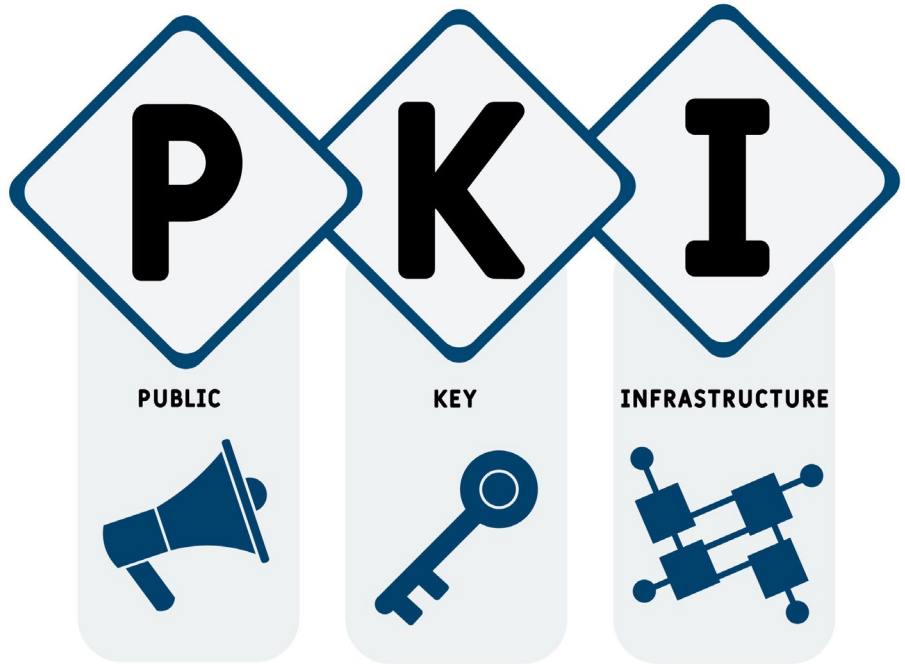
MÃ HOÁ HIỆN ĐẠI

CƠ SỞ HẠ TẦNG KHOÁ CÔNG KHAI
PUBLIC KEY INFRASTRUCTURE – PKI

Khái niệm:

Cơ sở hạ tầng khóa công khai (PKI) là công nghệ xác thực người dùng và thiết bị trong thế giới kỹ thuật số. Ý tưởng cơ bản là để một hoặc nhiều bên đáng tin cậy ký điện tử vào các tài liệu xác nhận rằng một khóa mật mã cụ thể thuộc về một người dùng hoặc thiết bị cụ thể. Sau đó, khóa có thể được sử dụng làm danh tính cho người dùng trong các mạng kỹ thuật số.

Người dùng và thiết bị có khóa thường chỉ được gọi là các thực thể. Nói chung, mọi thứ đều có thể được liên kết với một khóa mà nó có thể sử dụng làm danh tính của mình. Bên cạnh người dùng hoặc thiết bị, nó có thể là một chương trình, quy trình, nhà sản xuất, thành phần hoặc thứ gì đó khác. Mục đích của PKI là liên kết an toàn một khóa với một thực thể.



MÃ HOÁ HIỆN ĐẠI

CƠ SỞ HẠ TẦNG KHOÁ CÔNG KHAI
PUBLIC KEY INFRASTRUCTURE – PKI

Trong thế giới thực, có nhiều cơ quan cấp chứng chỉ và hầu hết các máy tính cũng như trình duyệt web đều tin tưởng khoảng một trăm cơ quan cấp chứng chỉ theo mặc định.

Cơ sở hạ tầng khóa công khai dựa trên công nghệ chữ ký số, sử dụng mật mã khóa công khai. Ý tưởng cơ bản là khóa bí mật của mỗi thực thể chỉ được biết bởi thực thể đó và được sử dụng để ký. Khóa này được gọi là khóa riêng. Có một khóa khác bắt nguồn từ nó, được gọi là khóa chung, được sử dụng để xác minh chữ ký nhưng không thể được sử dụng để ký. Khóa công khai này được cung cấp cho bất kỳ ai và thường được bao gồm trong tài liệu chứng chỉ.

MÃ HOÁ HIỆN ĐẠI

CƠ SỞ HẠ TẦNG KHOÁ CÔNG KHAI
PUBLIC KEY INFRASTRUCTURE – PKI

PKI cung cấp sự đảm bảo về khóa công khai. Nó cung cấp việc xác định các khóa công khai và phân phối chúng. Các thành phần của PKI bao gồm:

MÃ HOÁ HIỆN ĐẠI

CƠ SỞ HẠ TẦNG KHOÁ CÔNG KHAI
PUBLIC KEY INFRASTRUCTURE – PKI

Public Key Certificate – Chứng chỉ khoá công khai, hay còn được gọi là:
Digital Certificate - Chứng chỉ kỹ thuật số.

Giống như thực tế, chứng chỉ có thể được coi là thẻ căn cước được cấp cho người đó. Người dân sử dụng chứng minh thư như bằng lái xe, hộ chiếu để chứng minh danh tính. Chứng chỉ kỹ thuật số thực hiện cùng một điều cơ bản trong thế giới điện tử, nhưng có một điểm khác biệt.

Chứng chỉ số không chỉ được cấp cho con người mà chúng có thể được cấp cho máy tính, gói phần mềm hoặc bất kỳ thứ gì khác cần chứng minh danh tính trong thế giới điện tử.

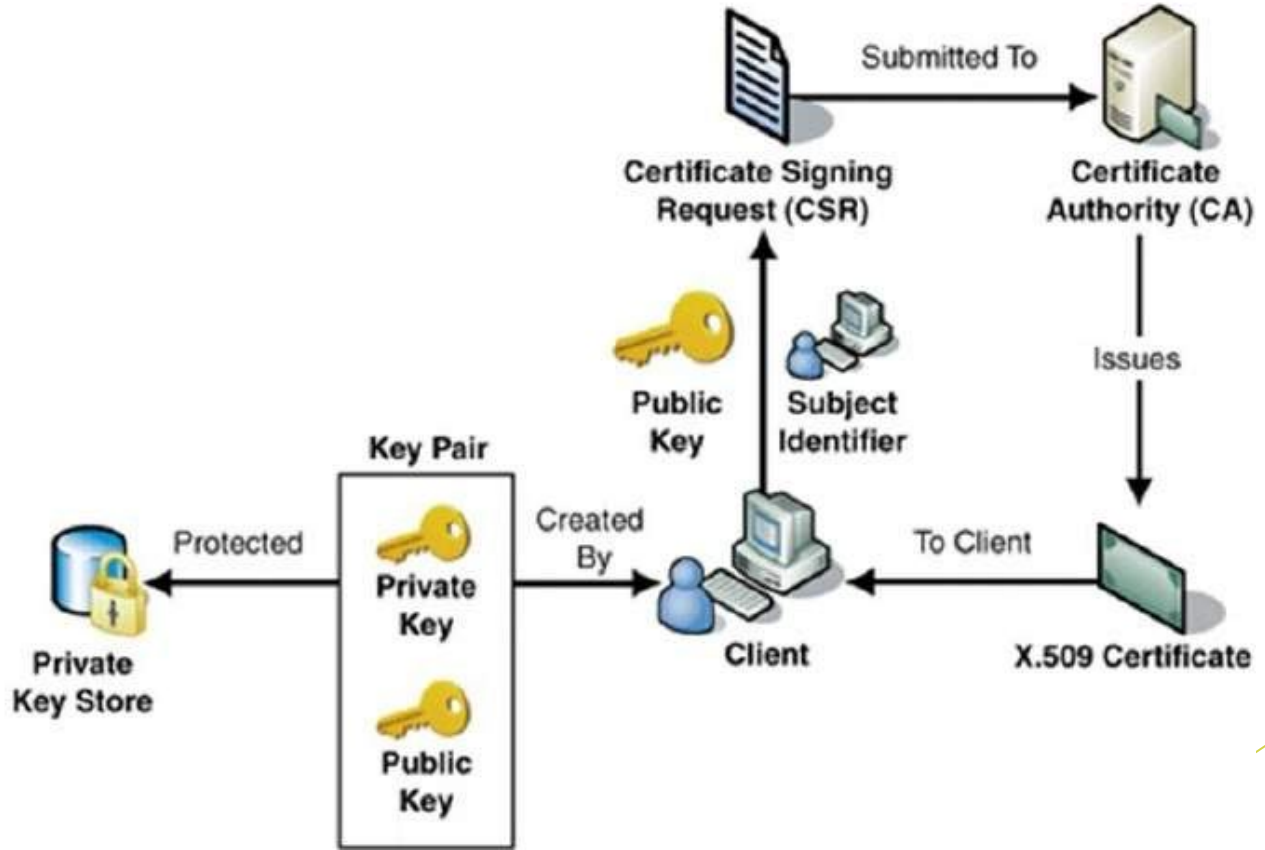
- Chứng chỉ kỹ thuật số dựa trên tiêu chuẩn ITU X.509 xác định định dạng chứng chỉ tiêu chuẩn cho chứng chỉ khóa công khai và xác thực chứng chỉ. Do đó, chứng chỉ kỹ thuật số đôi khi còn được gọi là chứng chỉ X.509.
- CA ký điện tử toàn bộ thông tin này và bao gồm chữ ký điện tử trong chứng chỉ.
- Bất kỳ ai cần sự đảm bảo về khóa chung và thông tin liên quan của khách hàng, anh ta sẽ thực hiện quy trình xác thực chữ ký bằng khóa chung của CA. Xác thực thành công đảm bảo rằng khóa công khai được cung cấp trong chứng chỉ thuộc về người có thông tin chi tiết được cung cấp trong chứng chỉ.

MÃ HOÁ HIỆN ĐẠI

CƠ SỞ HẠ TẦNG KHOÁ CÔNG KHAI
PUBLIC KEY INFRASTRUCTURE – PKI

Public Key Certificate – Chứng chỉ khoá công khai, hay còn được gọi là:
Digital Certificate - Chứng chỉ kỹ thuật số.

Quá trình lấy chứng chỉ kỹ thuật số cho một người/tổ chức được thực hiện theo lược đồ sau:



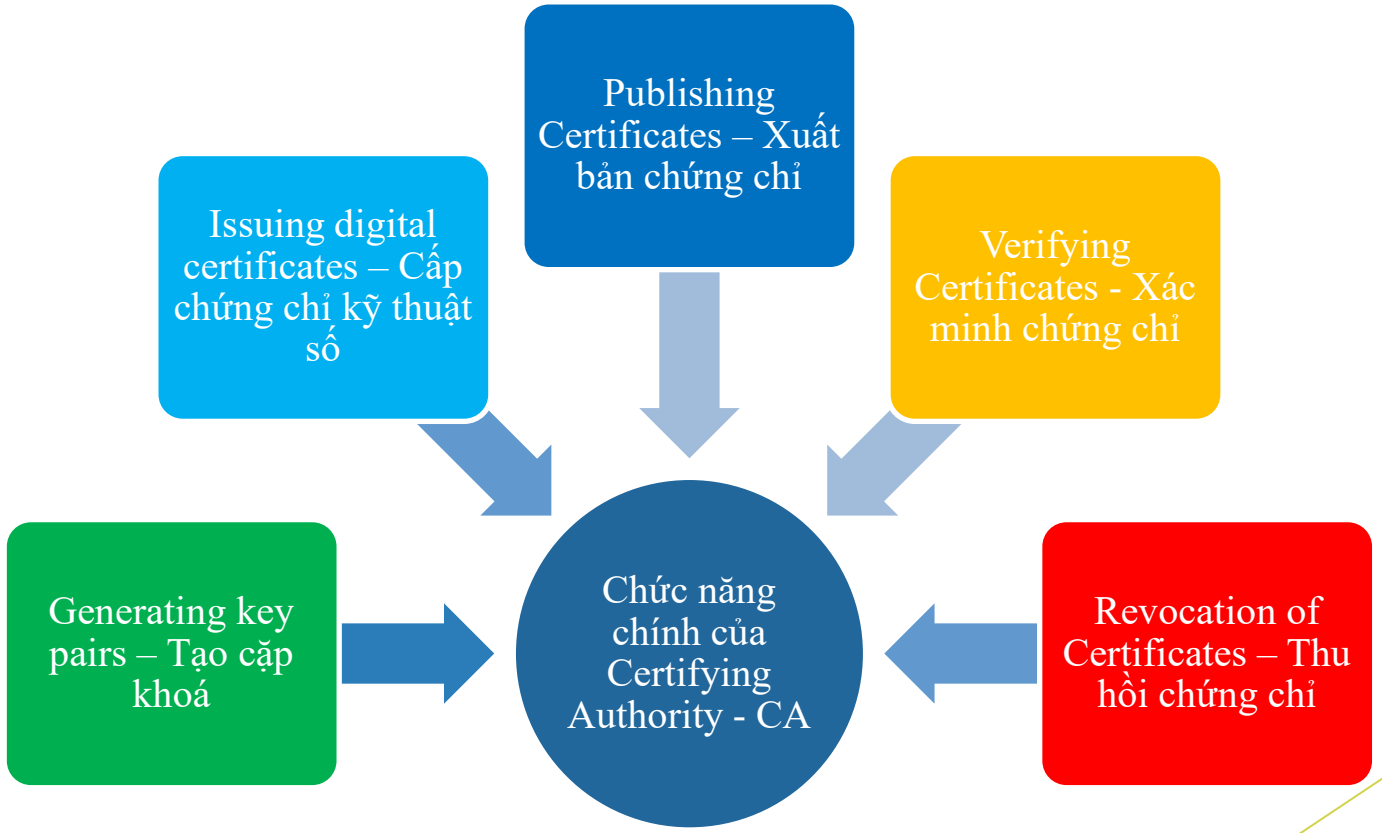
MÃ HOÁ HIỆN ĐẠI

CƠ SỞ HẠ TẦNG KHOÁ CÔNG KHAI
PUBLIC KEY INFRASTRUCTURE – PKI

Certifying Authority (CA)

Như đã đề cập, CA cấp chứng chỉ cho khách hàng và hỗ trợ những người dùng khác xác minh chứng chỉ. CA chịu trách nhiệm xác định chính xác danh tính của khách hàng yêu cầu cấp chứng chỉ và đảm bảo rằng thông tin có trong chứng chỉ là chính xác và ký điện tử vào chứng chỉ đó.

Các chức năng chính của CA:



MÃ HOÁ HIỆN ĐẠI

CƠ SỞ HẠ TẦNG KHOÁ CÔNG KHAI
PUBLIC KEY INFRASTRUCTURE – PKI

Certifying Authority (CA)

Có 4 loại chứng chỉ điển hình:

Loại 1
- Có thể dễ dàng nhận được các chứng chỉ này bằng cách cung cấp địa chỉ email.

Loại 2
- Các chứng chỉ này yêu cầu cung cấp thêm thông tin cá nhân.

Loại 3
- Chỉ có thể mua các chứng chỉ này sau khi đã kiểm tra danh tính của người yêu cầu.

Loại 4
- Chúng có thể được sử dụng bởi các chính phủ và tổ chức tài chính cần mức độ tin cậy rất cao.

MÃ HOÁ HIỆN ĐẠI

CƠ SỞ HẠ TẦNG KHOÁ CÔNG KHAI
PUBLIC KEY INFRASTRUCTURE – PKI

Registration Authority (RA)

CA có thể sử dụng Cơ quan đăng ký bên thứ ba (RA) để thực hiện các kiểm tra cần thiết đối với người hoặc công ty yêu cầu chứng chỉ để xác nhận danh tính của họ. RA có thể xuất hiện với khách hàng dưới dạng CA, nhưng họ không thực sự ký chứng chỉ được cấp.

Mục đích chính của RA là để giảm tải công việc của CA

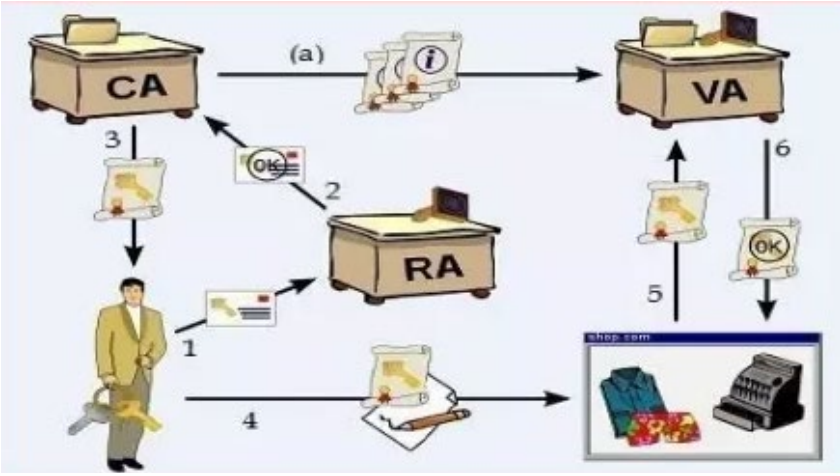
- Xác thực cá nhân, chủ thể đăng ký chứng thư số.
- Kiểm tra tính hợp lệ của thông tin do chủ thể cung cấp.
- Xác nhận quyền của chủ thể đối với những thuộc tính chứng thư số được yêu cầu.
- Kiểm tra xem chủ thể có thực sự sở hữu khóa riêng đang được đăng ký hay không (chứng minh sở hữu).
- Tạo cặp khóa bí mật, công khai. (nếu chủ thể yêu cầu)
- Phân phối bí mật được chia sẻ đến thực thể cuối (ví dụ khóa công khai của CA).
- Thay mặt chủ thể thực thể cuối khởi tạo quá trình đăng ký với CA.
- Lưu trữ khóa riêng.
- Khởi tạo quá trình khôi phục khóa
- Phân phối thẻ bài vật lý (thẻ thông minh)

MÃ HOÁ HIỆN ĐẠI

CƠ SỞ HẠ TẦNG KHOÁ CÔNG KHAI
PUBLIC KEY INFRASTRUCTURE – PKI

Certificate Repository – Kho lưu trữ chứng chỉ

- Hệ thống (có thể tập trung hoặc phân tán) lưu trữ chứng chỉ và danh sách các chứng chỉ bị thu hồi
- Cung cấp cơ chế phân phối chứng chỉ và danh sách thu hồi chứng chỉ (CRLs - Certificate Revocatio Lists).



- (1) : Người dùng gửi yêu cầu phát hành thẻ chứng chỉ số và khóa công khai đến RA(Registration Authority);
- (2) : Sau khi xác nhận tính hợp lệ định danh của người dùng thì RA sẽ chuyển yêu cầu này đến CA (Certifying Authority);
- (3) : CA phát hành thẻ chứng chỉ số cho người dùng;
- (4) : Sau đó người dùng "ký" thông điệp trao đổi với thẻ chứng chỉ số mới vừa nhận được từ CA và sử dụng chúng (thẻ chứng thực số + chữ ký số) trong giao dịch;
- (5) : Định danh của người dùng được kiểm tra bởi đối tác thông qua sự hỗ trợ của VA(Validation authority);
- (6) : Nếu chứng chỉ số của người dùng được xác nhận tính hợp lệ thì đối tác mới tin cậy người dùng và có thể bắt đầu quá trình trao đổi thông tin với nó (VA nhận thông tin về thẻ chứng chỉ số đã được phát hành từ CA)

MÃ HOÁ HIỆN ĐẠI

CƠ SỞ HẠ TẦNG KHOÁ CÔNG KHAI
PUBLIC KEY INFRASTRUCTURE – PKI

Private Key Tokens (RA)

Trong khi khóa công khai của khách hàng được lưu trữ trên chứng chỉ, khóa riêng tư bí mật được liên kết có thể được lưu trữ trên máy tính của chủ sở hữu khóa. Phương pháp này thường không được thông qua. Nếu kẻ tấn công giành được quyền truy cập vào máy tính, anh ta có thể dễ dàng giành được quyền truy cập vào khóa riêng. Vì lý do này, khóa riêng tư được lưu trữ trên thiết bị lưu trữ mã thông báo di động (USB Token), được bảo vệ an toàn thông qua mật khẩu.

Các nhà cung cấp khác nhau thường sử dụng các định dạng lưu trữ khác nhau và đôi khi là độc quyền để lưu trữ khóa.

Ví dụ: Entrust sử dụng định dạng .epf độc quyền, trong khi Verisign, GlobalSign và Baltimore sử dụng định dạng .p12 tiêu chuẩn.

MÃ HOÁ HIỆN ĐẠI

CƠ SỞ HẠ TẦNG KHOÁ CÔNG KHAI
PUBLIC KEY INFRASTRUCTURE – PKI

Hierarchy of CA – Phân cấp CA

Với các mạng rộng lớn và các yêu cầu về liên lạc toàn cầu, thực tế là không khả thi khi chỉ có một CA đáng tin cậy mà tất cả người dùng đều nhận được chứng chỉ đấy. Thứ hai, chỉ có một CA có thể dẫn đến khó khăn nếu CA bị xâm phạm.

Trong trường hợp như vậy, mô hình chứng nhận phân cấp được quan tâm vì nó cho phép sử dụng chứng chỉ khóa công khai trong môi trường mà hai bên giao tiếp không có mối quan hệ tin cậy với cùng một CA.

- CA gốc nằm ở trên cùng của hệ thống phân cấp CA và chứng chỉ của CA gốc là chứng chỉ tự ký.
- Các CA trực thuộc CA gốc (Ví dụ: CA1 và CA2) có chứng chỉ CA được ký bởi CA gốc.
- Các CA bên dưới các CA cấp dưới trong hệ thống phân cấp (Ví dụ: CA5 và CA6) có chứng chỉ CA của chúng được ký bởi các CA cấp dưới cấp cao hơn chúng.

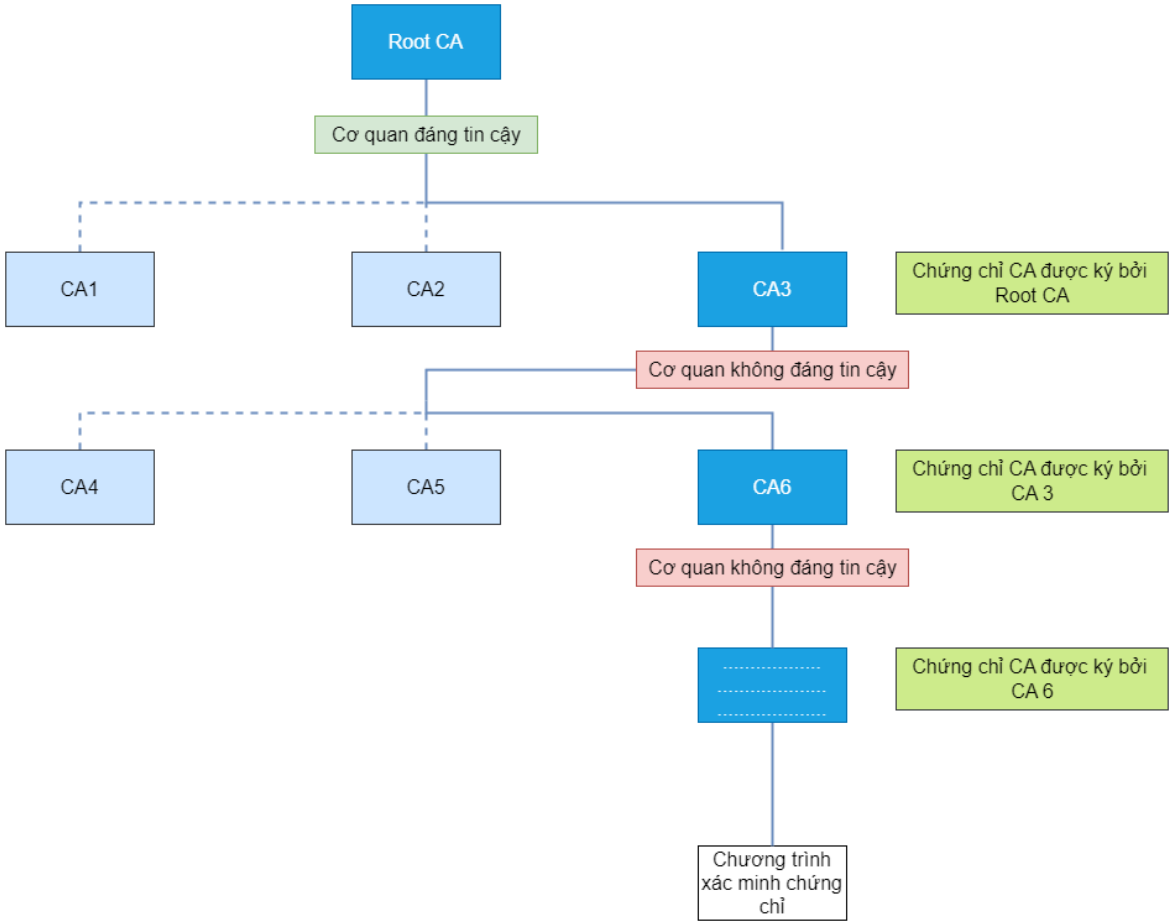
Hệ thống phân cấp của cơ quan cấp chứng chỉ (CA) được phản ánh trong các chuỗi chứng chỉ. Chuỗi chứng chỉ theo dõi đường dẫn chứng chỉ từ một nhánh trong hệ thống phân cấp đến thư mục gốc của hệ thống phân cấp.

MÃ HOÁ HIỆN ĐẠI

CƠ SỞ HẠ TẦNG KHOÁ CÔNG KHAI
PUBLIC KEY INFRASTRUCTURE – PKI

Hierarchy of CA – Phân cấp CA

Hình minh họa sau đây cho thấy cấu trúc phân cấp CA với chuỗi chứng chỉ dẫn từ chứng chỉ thực thể qua hai chứng chỉ CA cấp dưới (CA6 và CA3) đến chứng chỉ CA cho CA gốc.



MÃ HOÁ HIỆN ĐẠI

CƠ SỞ HẠ TẦNG KHOÁ CÔNG KHAI
PUBLIC KEY INFRASTRUCTURE – PKI

Hierarchy of CA – Phân cấp CA

Xác minh chuỗi chứng chỉ là quá trình đảm bảo rằng một chuỗi chứng chỉ cụ thể hợp lệ, được ký chính xác và đáng tin cậy. Quy trình sau đây xác minh chuỗi chứng chỉ, bắt đầu bằng chứng chỉ được xuất trình để xác thực:

- Máy khách có tính xác thực đang được xác minh cung cấp chứng chỉ của mình, thường cùng với chuỗi chứng chỉ cho đến Root CA.
- Người xác minh lấy chứng chỉ và xác thực bằng cách sử dụng khóa công khai của tổ chức phát hành. Khóa công khai của nhà phát hành được tìm thấy trong chứng chỉ của nhà phát hành nằm trong chuỗi bên cạnh chứng chỉ của khách hàng.
- Bây giờ nếu CA cấp cao hơn, người đã ký chứng chỉ của nhà phát hành, được người xác minh tin cậy, thì quá trình xác minh thành công và dừng tại đây.
- Mặt khác, chứng chỉ của tổ chức phát hành được xác minh theo cách tương tự như được thực hiện cho khách hàng trong các bước trên. Quá trình này tiếp tục cho đến khi CA đáng tin cậy được tìm thấy ở giữa hoặc nếu không thì nó sẽ tiếp tục cho đến CA gốc.

MÃ HOÁ HIỆN ĐẠI

CƠ SỞ HẠ TẦNG KHOÁ CÔNG KHAI
PUBLIC KEY INFRASTRUCTURE – PKI

Chứng nhận X.509

Chứng nhận X.509 là chứng nhận khóa công khai phổ biến nhất. Hiệp hội Viễn thông quốc tế (International Telecommunications Union – ITU) đã chỉ định chuẩn X.509 vào năm 1988. Đây là định dạng phiên bản 1 của chuẩn X.509. Vào năm 1993, phiên bản 2 của chuẩn X.509 được phát hành với 2 trường tên nhận dạng duy nhất được bổ sung. Phiên bản 3 của chuẩn X.509 được bổ sung thêm trường mở rộng đã phát hành vào năm 1997.

Một chứng nhận khóa công khai kết buộc một khóa công khai với sự nhận diện của một người (hoặc một thiết bị). Khóa công khai và tên thực thể sở hữu khóa này là hai mục quan trọng trong một chứng nhận.

MÃ HOÁ HIỆN ĐẠI

CƠ SỞ HẠ TẦNG KHOÁ CÔNG KHAI
PUBLIC KEY INFRASTRUCTURE – PKI

Chứng nhận X.509

- **Version:** Chỉ định phiên bản của chứng nhận X.509.
- **Serial Number:** Số loạt phát hành được gán bởi CA. Mỗi CA nên gán một mã số loạt duy nhất cho mỗi giấy chứng nhận mà nó phát hành.
- **Signature Algorithm:** Thuật toán chữ ký chỉ rõ thuật toán mã hóa được CA sử dụng để ký giấy chứng nhận. Trong chứng nhận X.509 thường là sự kết hợp giữa thuật toán băm (chẳng hạn như SHA256) và thuật toán khóa công khai (chẳng hạn như RSA).
- **Issuer Name:** Tên tổ chức CA phát hành giấy chứng nhận, đây là một tên phân biệt theo chuẩn X.500. Hai CA không được sử dụng cùng một tên phát hành.
- **Validity Period:** Trường này bao gồm 2 giá trị chỉ định khoảng thời gian mà giấy chứng nhận có hiệu lực.

  Hai phần của trường này là not-before và not-after.
  - **Not-before** chỉ định thời gian mà chứng nhận này bắt đầu có hiệu lực,
  - **Not-after** chỉ định thời gian mà chứng nhận hết hiệu lực.

Các giá trị thời gian này được đo theo chuẩn thời gian Quốc tế, chính xác đến từng giây.

MÃ HOÁ HIỆN ĐẠI

CƠ SỞ HẠ TẦNG KHOÁ CÔNG KHAI
PUBLIC KEY INFRASTRUCTURE – PKI

Chứng nhận X.509

- **Subject Name:** là một X.500 DN, xác định đối tượng sở hữu giấy chứng nhận mà cũng là sở hữu của khóa công khai. Một CA không thể phát hành 2 giấy chứng nhận có cùng một Subject Name.
- **Public key:** Xác định thuật toán của khóa công khai (như RSA) và chứa khóa công khai được định dạng tùy vào kiểu của nó.
- **Issuer Unique ID và Subject Unique ID:** Hai trường này được giới thiệu trong X.509 phiên bản 2, được dùng để xác định hai tổ chức CA hoặc hai chủ thể khi chúng có cùng DN. RFC 2459 đề nghị không nên sử dụng 2 trường này.
- **Extensions:** Chứa các thông tin bổ sung cần thiết mà người thao tác CA muốn đặt vào chứng nhận. Trường này được giới thiệu trong X.509 phiên bản 3.
- **Signature:** Đây là chữ ký điện tử được tổ chức CA áp dụng. Tổ chức CA sử dụng khóa bí mật có kiểu quy định trong trường thuật toán chữ ký. Chữ ký bao gồm tất cả các phần khác trong giấy chứng nhận. Do đó, tổ chức CA chứng nhận cho tất cả các thông tin khác trong giấy chứng nhận chứ không chỉ cho tên chủ thể và khóa công khai.

MÃ HOÁ HIỆN ĐẠI

CƠ SỞ HẠ TẦNG KHOÁ CÔNG KHAI
PUBLIC KEY INFRASTRUCTURE – PKI

Chứng nhận X.509

Những phần mở rộng của tên tập tin phổ biến cho chứng nhận X.509 bao gồm:

- **.cer:** chứng nhận được mã hóa theo luật mã hóa tiêu chuẩn (Canonical Encoding Rules – CER).
- **.der:** chứng nhận được mã hóa theo luật mã hóa phân biệt (Distinguished Encoding Rules – DER).
- **.pem (Privacy-Enhanced Electronic Mail):** định dạng mã hóa được sử dụng để lưu trữ các chứng nhận và khóa. Một tập tin được định dạng với chuẩn này có thể chứa các khóa bí mật (RSA và DSA), khóa công khai (RSA và DSA) và các chứng nhận X509. Định dạng này lưu trữ dữ liệu ở định dạng DER được mã hóa cơ sở 64, nằm giữa "-----BEGIN CERTIFICATE-----" và "-----END CERTIFICATE-----", phù hợp cho việc trao đổi ở dạng văn bản giữa các hệ thống.
- **.p7b, p7c:** PKCS #7 là một định dạng mã hóa cho việc lưu trữ một chứng nhận số và chuỗi chứng nhận của nó dưới dạng các ký tự ASCII. Định dạng này được sử dụng bởi CA để trả về các chứng nhận được phát hành cùng với chuỗi chứng nhận. Định dạng này có có thể được sử dụng như đầu vào cho yêu cầu gia hạn chứng nhận đến một CA.
- **.pfx, .p12:** PKCS #12 là một định dạng mã hóa cho việc lưu trữ một chứng nhận số và kết hợp với khóa bí mật dưới dạng các ký tự ASCII. Định dạng này luôn luôn được trả về bởi CA khi CA phát sinh các khóa và phát hành chứng nhận đồng thời.