MỤC LỤC

DANH MỤC HÌNH VẼ

LỜI CẢM ƠN

Trong lời đầu tiên của báo cáo Đồ án Tốt Nghiệp "Tìm hiểu Hệ thống cung cấp chứng chỉ số theo mô hình sinh khóa tập trung" này, em muốn gửi lời cám ơn và biết ơn chân thành nhất của mình tới tất cả những người đã hỗ trợ, giúp đỡ em về kiến thức và tinh thần trong quá trình thực hiện Đồ án.

Trước hết, em xin chân thành cảm ơn Thày Giáo – TS. Hồ Văn Canh, Cố vấn cục kĩ thuật nghiệp vụ 1-Bộ CA, người đã trực tiếp hướng dẫn, nhận xét, giúp đỡ em trong suốt quá trình thực hiện Đồ án. Xin chân thành cảm ơn GS.TS.NGƯT Trần Hữu Nghị Hiệu trưởng Trường Đại học Dân lập Hải Phòng, ban giám hiệu nhà trường, các thày cô trong Khoa Công Nghệ Thông Tin và các phòng ban nhà trường đã tạo điều kiện tốt nhất cho em cũng như các bạn khác trong suốt thời gian học tập và làm tốt mnghiệp.

Cuối cùng em xin gửi lời cảm ơn đến gia đình, bạn bè, người thân đã giúp đỡ động viên em rất nhiều trong quá trình học tập và làm Đồ án Tốt Nghiệp.

Do thời gian có hạn, kiến thức còn nhiều hạn chế nên Đồ án thực hiện chắc chắn không tránh khỏi những thiếu sót nhất định. Em rất mong nhận được ý kiến đóng góp của thày cô và các bạn để em có thêm kinh nghiệm và tiếp tục hoàn thiện Đồ án của mình.

Em xin chân thành cảm ơn!

Hải Phòng, ngày tháng năm 2012

Sinh viên

Nguyễn Tiến Hoàng

MỞ ĐẦU

Trong một vài năm lại đây, hạ tầng truyền thông IT càng ngày càng được mở rộng khi người sử dụng dựa trên nền tảng này để truyền thông và giao dịch với các đồng nghiệp, các đối tác kinh doanh cũng như việc khách hàng dùng email trên các mạng công cộng. Hầu hết các thông tin nhạy cảm và quan trọng được lưu trữ và trao đổi dưới hình thức điện tử trong các cơ quan văn phòng, doanh nghiệp. Sự thay đổi trong các hoạt động truyền thông này đồng nghĩa với việc cần phải có biện pháp bảo vệ đơn vị, tổ chức, doanh nghiệp của mình trước các nguy cơ lừa đảo, can thiệp, tấn công, phá hoại hoặc vô tình tiết lộ các thông tin đó. Cơ sở hạ tầng khóa công khai (PKI – Public Key Infrastructure) cùng các tiêu chuẩn công nghệ ứng dụng của nó có thể coi là một giải pháp tổng hợp và độc lập có thể sử dụng để giải quyết vấn đề này. PKI bản chất là một hệ thống công nghệ vừa mang tính tiêu chuẩn, vừa mang tính ứng dụng được sử dụng để khởi tạo, lưu trữ và quản lý các chứng chỉ số hay ta còn gọi là chức thực điện tử (digital certificate) cũng như các khóa công cộng (khóa công khai) và cá nhân (khóa riêng). Sáng kiến PKI ra đời năm 1995, khi mà các chính phủ và các tổ chức công nghiệp xây dựng các tiêu chuẩn chung dựa trên phương pháp mã hóa để hỗ trợ một hạ tầng bảo mật trên mạng Internet. Tại thời điểm đó, mục tiêu đước đặt ra là xây dựng một bộ tiêu chuẩn bảo mật tổng hợp cùng các công cụ và lý thuyết cho phép người sử dụng cũng như các tổ chức có thể tạo lập, lữu trữ và trao đổi các thông tin một cách an toàn trong phạm vi cá nhân và công cộng.

Hiện nay ở Việt Nam, việc nghiên cứu, ứng dụng và triển khai PKI nói chung và dịch vụ cung cấp chứng chỉ số nói riêng là vấn đề còn mang tính thời sự. Bằng việc sử dụng chứng chỉ và chữ ký số, những ứng dụng cho phép PKI đưa ra nhiều đặc tính đảm bảo an toàn thông tin cho người sử dụng. Có hai mô hình cung cấp chứng chỉ số là mô hình do CA sinh cặp khóa công khai và khóa bí mật cho người dùng và mô hình do tự người dùng sinh cặp khóa công khai và khóa bí mật cho chính mình. Hiện nay, ở Việt Nam đang nghiên cứu và triển khai hệ thống PKI theo mô hình thứ nhất. Vì vậy em chọn đề tài "Tìm hiểu Hệ thống cung cấp chứng chỉ số theo mô hình sinh khóa tập trung" để làm đề tài đồ án tốt nghiệp.

Chương 1: CÁC THÀNH PHẦN KỸ THUẬT CƠ BẢN TRONG PKI (PUBLIC KEY INFRASTRUCTURE)

Mã hóa là công cụ cơ bản của việc đảm bảo an toàn dữ liệu. Ở thời kỳ sơ khai, con người đã sử dụng nhiều phương pháp để bảo vệ các thông tin bí mật, nhưng tất cả các phương pháp đó chỉ mang tính nghệ thuật hơn là khoa học. Ban đầu, mật mã học được sử dụng phổ biến cho quân đội, qua nhiều cuộc chiến tranh, vai trò của mật mã ngày càng quan trọng và mang lại nhiều thành quả không nhỏ như các hệ mã cổ điển Caeser, Playfair,…Chúng đã là nền tảng cho mật mã học này nay.

Ngày nay, khi toán học được áp dụng cho mật mã học thì lịch sử của mật mã học đã sang trang mới. Việc ra đời các hệ mã hóa đối xứng không làm mất đi vai trò của các hệ mật mã cổ điển mà còn bổ sung cho ngành mật mã nhiều phương pháp mã hóa mới. Từ năm 1976, khi hệ mật mã phi đối xứng (mật mã khóa công khai) ra đời, nhiều khái niệm mới gắn với mật mã học đã xuất hiện: chữ ký số, hàm băm, mã đại diện, chứng chỉ số. Mật mã học không chỉ áp dụng cho quân sự mà còn cho các lĩnh vực kinh tế xã hội khác như giao dịch hành chính, thương mại điện tử.

Hiện nay có nhiều phương pháp mã hóa khác nhau, mỗi phương pháp có ưu, nhược điểm riêng. Tùy theo yêu cầu của môi trường ứng dụng nào, người ta có thể dùng phương pháp này hay phương pháp kia. Có những môi trường cần phải an toàn tuyệt đối bất kể thời gian và chi phí. Có những môi trường lại cần giải pháp dung hòa giữa bảo mật và chi phí.

Các thông điệp cần chuyển đi và cần được bảo vệ an toàn gọi là bản rõ (plaintext), và được ký hiệu là P. Nó có thể là một dòng các bít, các file, âm thanh số hoá,... Bản rõ được dùng để lưu trữ hoặc để truyền đạt thông tin. Trong mọi trường hợp bản rõ là thông điệp cần mã hoá. Quá trình xử lý một thông điệp trước khi gửi được gọi là quá trình mã hoá (encryption). Một thông điệp đã được mã hoá được gọi là bản mã (ciphertext), và được ký hiệu là C. Quá trình xử lý ngược lại từ bản mã thành bản rõ được gọi là quá trình giải mã (decryption).

Để bảo đảm an toàn thông tin lưu trữ trong máy tính (Ví dụ giữ gìn thông tin cố định) hay bảo đảm an toàn thông tin trên đường truyền tin (Ví dụ trên mạng máy tính, trên điện thoại), người ta phải “Che Giấu” các thông tin này.

"Che" thông tin (dữ liệu) hay "Mã hóa " thông tin là thay đổi hình dạng thông tin gốc, và người khác "khó" nhận ra.

"Giấu" thông tin (dữ liệu) là cất giấu thông tin trong bản tin khác, và người khác cũng "khó" nhận ra.

Trong phần này chúng ta bàn về "Mã hóa" thông tin

Hệ mật mã là tập hợp các thuật toán, các khóa nhằm che dấu thông tin tin cũng như làm rõ nó.

Hệ mật mã được định nghĩa là bộ năm (P,C,K,E,D), trong đó:

- P là tập hữu hạn các bản rõ có thể
- C là tập hữu hạn các bản mã có thể
- K là tập hữu hạn khóa có thể
- E là tập các hàm lập mã
- D là tập các hàm giải mã. Với mỗi k∈K có một hàm lập mã E_k∈E (E_k: P→ C) và một hàm giải mã D_k∈D (D_k: C → P) sao cho Dk (Ek (x)) = x , ∀ x ∈P.

Với khóa lập mã k_1 ∈ K, có hàm lập mã E_{k1}∈ E, E_{k1}: P→ C,

Với khóa giải mã k_2∈ K, có hàm giải mã D_{k2}∈ D,D_{k2}: C→ P,

sao cho D_{k2} (E_{k1}(x)) = x, ∀ x ∈ P.

Ở đây x được gọi là bản rõ, E_{k1}(x) = y được gọi là bản mã.

Trên đường truyền tin, thông tin được mã hoá để bảo đảm bí mật:

Người gửi G → → E_{k1}(T) → → Người nhận N

↑

Tin tặc có thể trộm bản mã E_{k1}(T) nhưng không có khóa $k_2$nên khó có thể giải mã để tìm ra bản rõ T.

Người gửi G muốn gửi bản tin T cho người nhận N. Để bảo đảm bí mật, G mã hoá bản tin bằng khóa lập mã k_1, nhận được bản mã E_{k1}(T), sau đó gửi cho N.

Tin tặc có thể trộm bản mã E_{k1}(T), nhưng "khó" tìm được bản tin gốc T nếu không có khoá giải mã k_2.

Người N nhận được bản mã, họ dùng khoá giải mã k_2, giải bản mã E_{k1}(T), để nhận được bản tin gốc T = D_{k2}(E_{k1}(T)).

Hiện nay các hệ mật mã được phân làm hai loại chính là: Hệ mật mã đối xứng và hệ mật mã bất đối xứng (hay còn gọi là hệ mật mã khóa công khai).

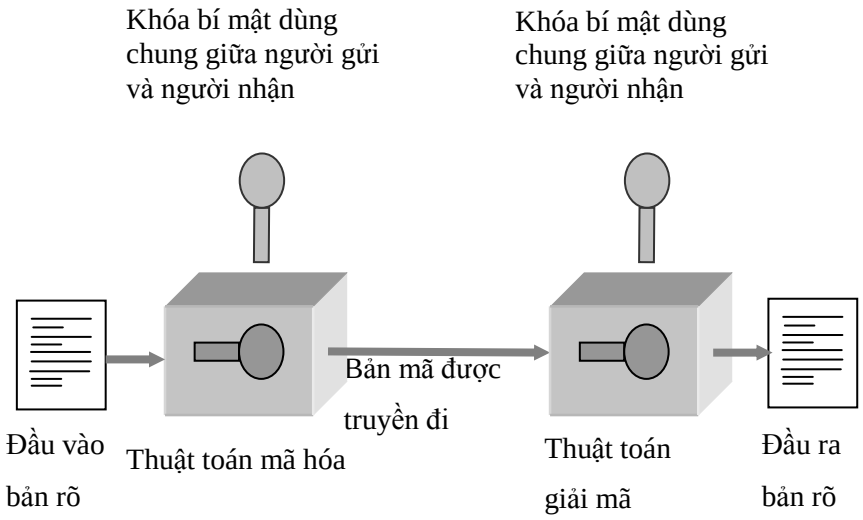
Mật mã đối xứng: có khóa lập mã và khóa giải mã "giống nhau", theo nghĩa biết được khóa này thì "dễ" tính được khóa kia. Phải giữ bí mật cả 2 khóa. Các hệ mật mã đối xứng như: Caesar, IDEA, DES, Triple DES.

Mật mã khóa công khai: có khóa lập mã khác khóa giải mã ($k_1 \neq k_2$), biết được khóa này cũng "khó" tính được khóa kia. Bí mật khóa giải mã. Công khai khóa lập mã. Các hệ mật mã khóa công khai RSA, Elgamal, ECC.

1.1 Hệ mã hóa khóa đối xứng

Mã hóa khóa đối xứng là Hệ mã hóa có khóa lập mã và khóa giải mã "giống nhau", theo nghĩa biết được khóa này thì "dễ" tính được khóa kia. Đặc biệt một số Hệ mã hóa loại này có khoá lập mã và khoá giải mã trùng nhau (k_1=k_2).

Hệ mã hóa khóa đối xứng còn có tên gọi là Hệ mã hóa khoá bí mật, vì phải giữ bí mật cả 2 khóa. Trước khi dùng Hệ mã hóa khóa đối xứng, người gửi và người nhận phải thoả thuận thuật toán mã hóa và một khoá chung (lập mã hay giải mã), khoá này phải được giữ bí mật. Độ an toàn của Hệ mã hóa loại này phụ thuộc vào khoá.



Hình 1.1.1 Mô hình mã hóa đối xứng

Khóa bí mật dùng chung giữa người gửi và người nhận nếu được sinh ra bởi người gửi (hoặc người gửi), khóa phải được chuyển cho người còn lại theo một kênh

bí mật nào đó. Có thể dùng một thành viên thứ 3 (đáng tin cậy) sinh khóa và phân phối khóa một cách bí mật cho cả người gửi và người nhận.

1.1.1 Đặc điểm của hệ mã hóa khóa đối xứng

❖ Ưu điểm:

- Tốc độ mã hóa và giải mã nhanh.
- Sử dụng đơn giản: chỉ cần dùng một khoá cho cả 2 bước mã và giải mã.

❖ Nhược điểm:

- Mã hóa khóa đối xứng chưa thật an toàn với lý do đơn giản: Người mã hoá và người giải mã phải có "chung" một khoá. Khóa phải được giữ bí mật tuyệt đối, vì "dễ" xác định khoá này nếu biết khoá kia. Do đó, việc xác thực thông điệp và ký số là rất khó thực hiện.
- Khi hai người (lập mã, giải mã) cùng biết "chung" một bí mật, thì khó giữ được bí mật !
- Vấn đề thỏa thuận khoá và quản lý khóa chung là khó khăn và phức tạp. Người gửi và người nhận phải luôn thống nhất với nhau về khoá. Việc thay đổi khoá là rất khó và dễ bị lộ. Khóa chung phải được gửi cho nhau trên kênh an toàn.

1.1.2 Nơi sử dụng hệ mã hóa khóa đối xứng

Hệ mã hóa khóa đối xứng thường được sử dụng trong môi trường mà khoá chung có thể dễ dàng trao chuyển bí mật, chẳng hạn trong cùng một mạng nội bộ.

Hệ mã hóa khóa đối xứng dùng để mã hóa những bản tin lớn, vì tốc độ mã hóa và giải mã nhanh hơn Hệ mã hóa khóa công khai.

1.2 Hệ mã hóa khóa công khai

Khái niệm mật mã khoá công khai nảy sinh khi giải quyết hai vấn đề khó khăn trong mã hóa đối xứng.

Vấn đề đầu tiên là phân phối khoá. Như chúng ta đã biết, việc phân phối khoá trong mã hoá đối xứng yêu cầu hai bên liên lạc:

- Dùng chung một khoá được phân phối theo cách nào đó; hoặc:
- Sử dụng một trung tâm phân phối khoá.

Whitfield Diffie, một trong những người đã phát minh ra mã hoá khoá công khai (cùng với Martin Hellman, trường Đại học Stanford) đã suy luận và cho rằng, yêu cầu thứ hai phủ nhận bản chất của mật mã. Bản chất đó là đảm bảo tính bí mật trong liên lạc. Khó có thể tồn tại các hệ thống mật mã không thể phá được, nếu người sử dụng của các hệ thống này bắt buộc phải dùng chung các khoá của một trung tâm phân phối khoá (KDC), lý do là trung tâm này có thể để lộ khoá.

Vấn đề thứ hai mà Diffie đặt ra là "chữ ký số". Nếu việc sử dụng mật mã trở nên phổ biến, không chỉ trong lĩnh vực quân sự mà còn được sử dụng cho các mục đích thương mại và cá nhân, thì các thông báo và tài liệu điện tử cần có các chữ ký và chúng có hiệu lực tương tự như các chữ ký trên giấy tờ.

Các thuật toán khoá công khai sử dụng một khoá để mã hoá và một khoá khác để giải mã (tạo thành một cặp khoá). Chúng có tính chất quan trọng sau đây: “Khó có thể xác định được khoá giải mã nếu chỉ căn cứ vào các thông tin về thuật toán và khoá mã hoá.”

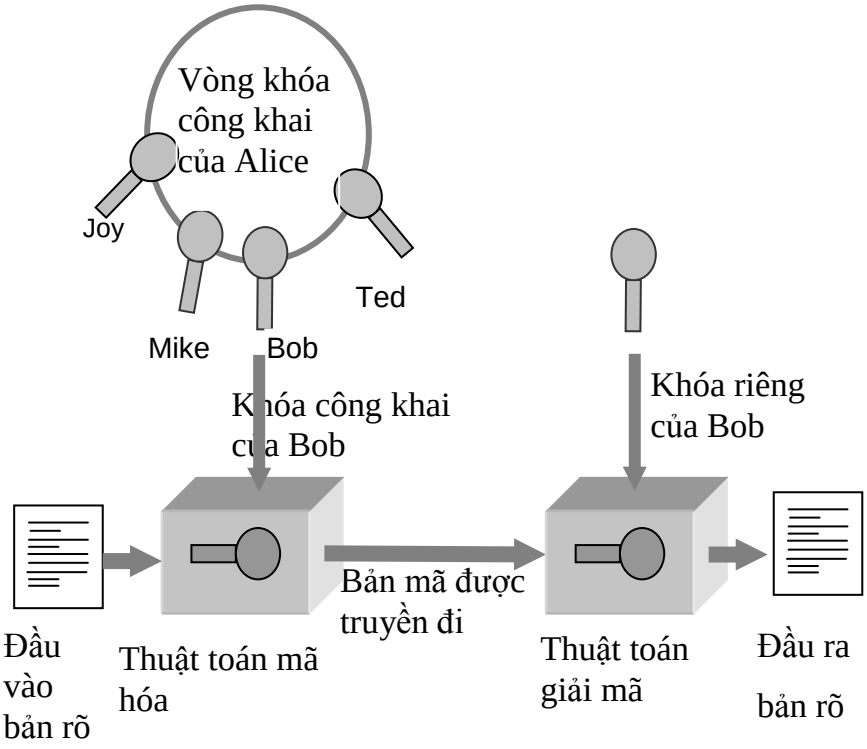
Mã hóa khóa công khai hay còn gọi mã hóa khóa phi đối xứng là Hệ mã hóa có khóa lập mã và khóa giải mã khác nhau ($k_1 \neq k_2$), biết được khóa này cũng “khó” tính được khóa kia.

Hệ mã hóa này còn được gọi là Hệ mã hoá khóa công khai, vì:

Khoá lập mã cho công khai, gọi là khoá công khai (Public key).

Khóa giải mã giữ bí mật, còn gọi là khóa riêng (Private key).

Một người bất kỳ có thể dùng khoá công khai để mã hoá bản tin, nhưng chỉ người nào có đúng khoá giải mã thì mới có khả năng xem được bản rõ.



Hình 1.1.2 Mô hình mã hóa khóa công

Hình trên minh hoạ quá trình mã hoá khoá công khai. Các bước cơ bản gồm:

- Mỗi hệ thống trên một mạng sinh ra một cặp khóa, cặp khoá này được sử dụng để mã hoá và giải mã các thông báo mà nó nhận được.
- Mỗi hệ thống công bố khóa mã hoá của mình bằng cách đặt khoá này vào trong một thanh ghi công khai hoặc một file. Đây chính là khoá công khai. Khoá cùng cặp được giữ bí mật.
- Nếu A muốn gửi cho B một thông báo, nó mã hoá thông báo bằng khoá công khai của B.
- Khi B nhận được thông báo, B giải mã thông báo bằng khoá riêng của B. Không một người nhận nào khác có thể giải mã thông báo, bởi vì chỉ có B mới biết khoá riêng của mình.

Với cách giải quyết này, tất cả các thành viên tham gia truyền thông có thể truy nhập vào các khoá công khai. Khoá riêng do mỗi thành viên sinh ra không bao giờ được phân phối. Quá trình liên lạc chỉ an toàn chừng nào hệ thống còn kiểm soát được khoá riêng của mình. Một hệ thống có thể thay đổi các khoá riêng của nó bất cứ lúc nào, đồng thời công bố các khoá công khai cùng cặp để thay thế khoá công khai cũ.

1.2.1 Đặc điểm cả hệ mã hóa công khai

❖ Ưu điểm:

o Người mã hoá dùng khóa công khai, người giải mã giữ khóa bí mật. Khả năng lộ khóa bí mật khó hơn vì chỉ có một người gìn giữ.

o Nếu kẻ phá hoại biết khoá công khai, cố gắng tìm khoá bí mật, thì chúng phải đương đầu với bài toán "khó".

o Khi biết các tham số ban đầu của hệ mã hóa, việc tính ra cặp khoá công khai và bí mật phải là "dễ", tức là trong thời gian đa thức.

o Người gửi có bản rõ P và khoá công khai, thì "dễ" tạo ra bản mã C.

o Người nhận có bản mã C và khoá bí mật, thì "dễ" giải được thành bản rõ P.

o Nếu kẻ phá hoại biết khoá công khai và bản mã C, thì việc tìm ra bản rõ P cũng là bài toán "khó", số phép thử là vô cùng lớn, không khả thi.

o Hệ mã hóa khóa công khai tiện lợi hơn Hệ mã hóa đối xứng cổ điển còn ở chỗ:

o Thuật toán được viết một lần, công khai cho nhiều lần dùng và cho nhiều người dùng, chỉ cần giữ bí mật khóa riêng.

❖ Nhược điểm:

o Mã hóa khóa công khai mã hóa và giải mã chậm hơn Mã hóa khóa đối xứng.

1.2.2 Nơi sử dụng hệ mã hóa công khai

Sử dụng chủ yếu trên các mạng công khai như Internet, khi mà việc trao đổi khoá bí mật tương đối khó khăn. Đặc trưng nổi bật của hệ mã hoá công khai là cả khoá công khai (public key) và bản mã (ciphertext) đều có thể gửi đi trên một kênh truyền tin không an toàn.

1.3 Công nghệ OpenCA

OpenCA là dự án đồ sộ nằm trong dự án OpenCA Group, có mục đích xây dựng PKI hoàn chỉnh, chuyên nghiệp, OpenCA được phát triển liên tục từ năm 1999. Từ năm 2001, OpenCA đã bắt đầu được sử dụng cho các đơn vị cỡ vừa và lớn.

OpenCA sử dụng giao diện web, hỗ trợ hầu hết các web Browser chính, hỗ trợ sản phẩm mã nguồn mở.

❖ Các Module chương trình trong OpenCA.

• Giao tiếp công cộng: Giao diện web để người sử dụng có thể truy cập qua Internet. Người dùng có thể đăng ký xin cấp chứng chỉ trực tiếp qua Module này.

• Giao tiếp LDAP: Danh bạ các khoá công khai, người dùng lấy khoá công khai từ Module này để mã hoá tài liệu, trước khi gửi đến đơn vị dùng openCA.

• Giao tiếp RA: Đơn vị điều hành RA sử dụng Module này để cập nhật các thông tin cá nhân của người xin cấp chứng chỉ.

• Giao tiếp OCSP: Module hỗ trợ kiểm tra chứng chỉ còn hiệu lực hay không. OCSP có tác dụng như việc công bố CRL, nhưng tính năng ưu việt hơn CRL.

• Giao tiếp CA: Module ký số riêng rẽ, cho phép CA làm theo nguyên tắc an ninh - tách biệt khỏi mạng công cộng, để bảo vệ tối đa khoá bí mật. Điều này khiến cho openCA trở nên an toàn hơn các phần mềm CA khác có trên thị trường hiện nay.

❖ Ngoài tính năng thiết yếu của PKI, OpenCA có nhiều tính năng ưu việt khác như:

• Đăng nhập bằng chứng chỉ.

• Hệ thống quản lý mềm dẻo.

• Sử dụng được các tính năng của X.509 mở rộng.

• OpenCA là phần mềm mã nguồn mở miễn phí, có tài liệu chi tiết đầy đủ.

OpenCA được thiết kế cho một hạ tầng phân tán. Nó có thể không chỉ điều khiển một CA offline và một RA online, mà còn giúp ta xây dựng một cấu trúc thứ bậc với nhiều mức khác nhau. OpenCA không phải là một giải pháp nhỏ cho các nghiên cứu vừa và nhỏ. Nó hỗ trợ tối đa cho các tổ chức lớn như các trường đại học, các công ty lớn.

1.3.1 Thiết kế tổng quan

OpenCA được thiết kế cho một hạ tầng phân tán. Nó có thể không chỉ điều khiển một CA offline và một RA online mà việc sử dụng chúng còn giúp ta xây dựng một cấu trúc thứ bậc với nhiều mức khác nhau. OpenCA không phải là một giải pháp

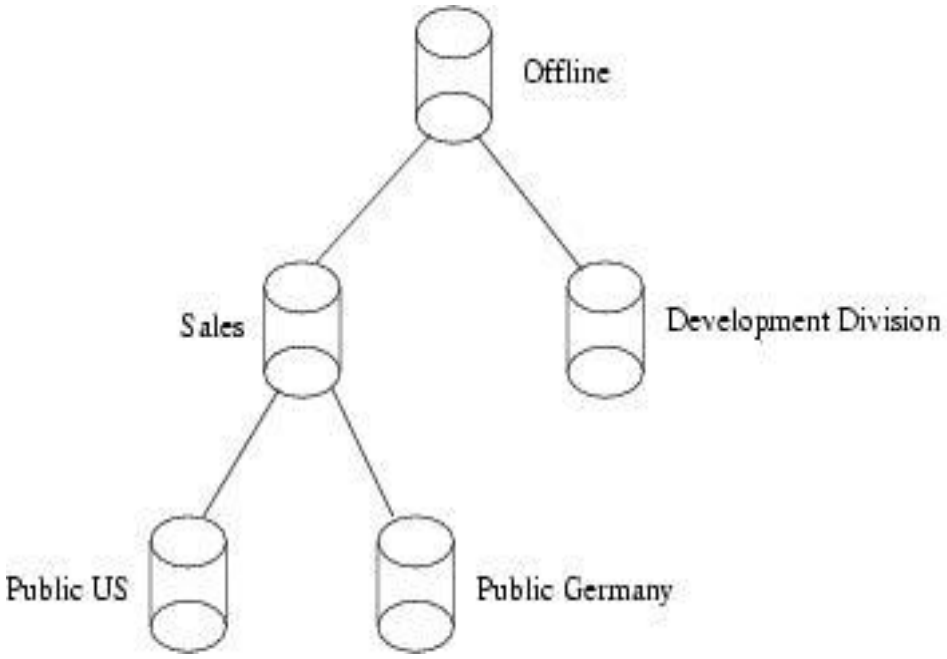
nhỏ cho các nghiên cứu vừa và nhỏ. Nó hỗ trợ tối đa cho các tổ chức lớn như các trường đại học, các công ty lớn.

Khi nghiên cứu về công nghệ OpenCA, chúng ta sẽ xem xét 4 phần chính:

- Thiết kế để cài đặt một hạ tầng tố
- Các hoạt động được thực hiện một cách offline bởi người quản trị
- Các thao tác phía người dùng
- Các mô tả kỹ thuật của OpenCA

1.3.2 Hệ thống thứ bậc

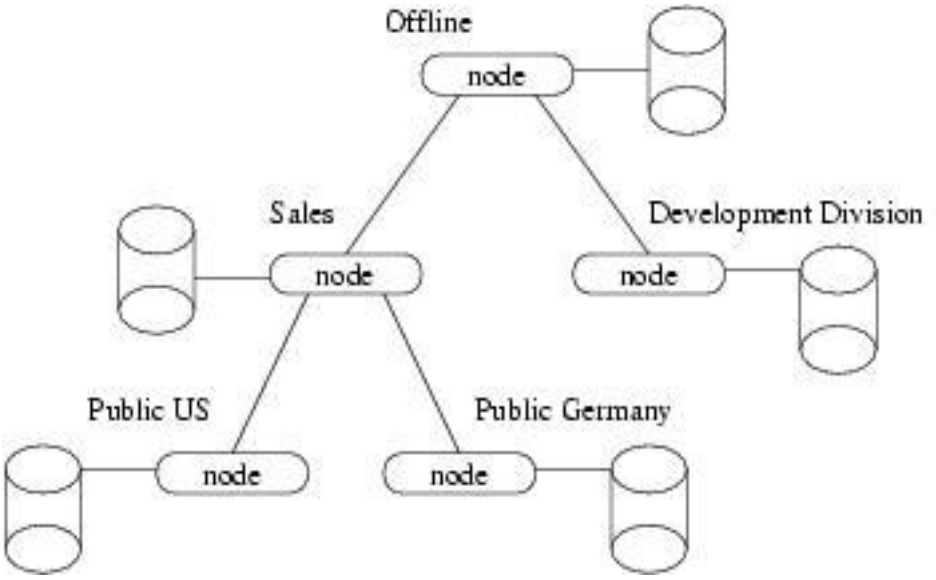
Kiến trúc cơ bản của các PKI X.509 là cấu trúc thứ bậc. Kết quả là chúng ta có cấu trúc cây cho các cơ sở dữ liệu nếu muốn tạo kiến trúc PKI phân tán.



Hình 1.3.2.1 Cái nhìn hướng CSDL của PKI

Trao đổi dữ liệu giữa các cơ sở dữ liệu (CSDL) riêng biệt có thể được kiểm soát tự động nếu ta sử dụng hệ thống CSDL phân tán. Nếu có CSDL phân tán (chẳng hạn một CA offline) thì ta phải có công nghệ cho việc trao đổi và quản lý các Node trong hệ thống cấp bậc.

Thiết kế của một OpenCA như sau:



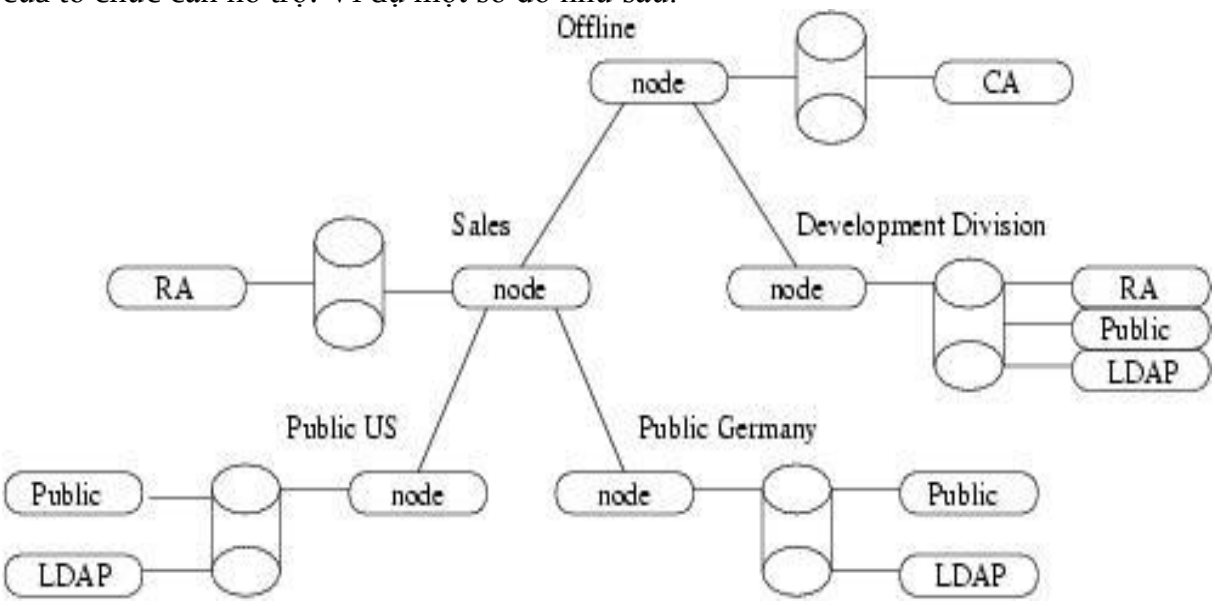
Hình 1.3.2.2 Cái nhìn dữ liệu logic

Thông thường mỗi Server trong hạ tầng của trung tâm tin cậy có CSDL riêng vì lý do an ninh. Cấu trúc thứ bậc là xương sống của trung tâm tin cậy.

1.3.3 Các giao diện

Sau khi biết hạ tầng cơ bản của OpenCA, Ta có thể tìm hiểu về CA, RA, LDAP và giao diện chung. OpenCA hỗ trợ tất cả các phần mềm qua giao diện Web.

Muốn thiết kế một trung tâm tin cậy mạnh, phải hình dung ra luồng công việc của tổ chức cần hỗ trợ. Ví dụ một sơ đồ như sau:



Hình 1.3.3.1 Cái nhìn kỹ thuật của PKI

OpenCA hỗ trợ các giao diện sau:

Node (chỉ cho Node quản lý), CA, RA, LDAP, Pub, SCEP

a. Node

CA có thể tạo tất cả các bảng, nhưng tự CA không thực hiện việc này. Giao diện cho các Node phục vụ việc back up và khôi phục khóa riêng của CA, chứng chỉ. CA không có cơ chế mặc định để thực hiện việc này mà chúng ta phải tự thực hiện qua giao diện được cung cấp.

b. CA

Giao diện CA có các chức năng để tạo các chứng chỉ và các danh sách thu hồi chứng chỉ. CA cũng cho chức năng để thay đổi hoặc cấu hình lại các giao diện.

c. RA

RA của OpenCA có khả năng điều khiển các loại yêu cầu: soạn thảo, chấp thuận, tạo các khóa riêng trên Smart card, xóa các yêu cầu không hợp lệ.

d. LDAP

Giao diện LDAP giúp cho việc tách biệt quản lý LDAP và phần còn lại của phần mềm. Điều này là cần thiết vì có nhiều chức năng chỉ cần cho người quản trị mà người dùng hoàn toàn không cần.

e. Pub

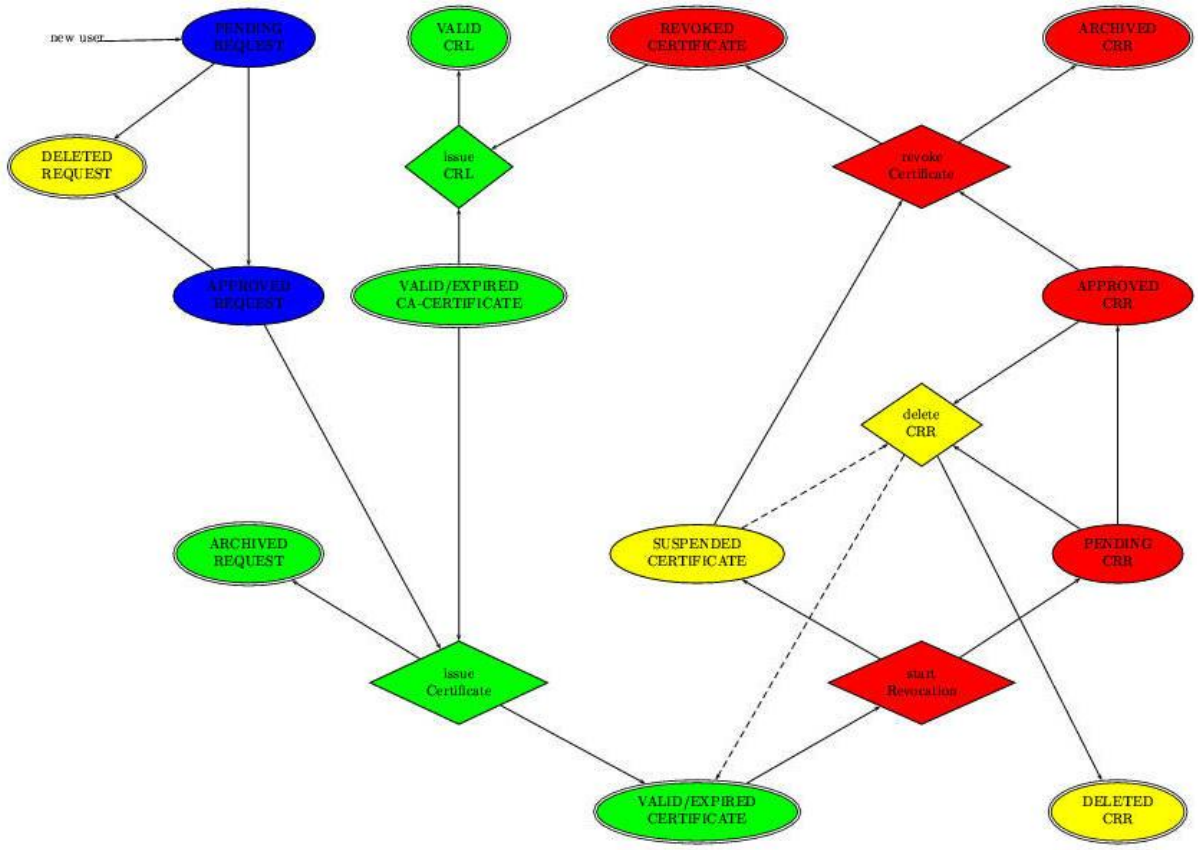
Giao diện công khai bao gồm những thứ liên quan đến người dùng.

- Sinh các CSRs (yêu cầu ký chứng chỉ) cho IE
- Sinh các CSRs (yêu cầu ký chứng chỉ) cho Mozilla 1.1+ and Netscape Communicator and Navigator
- Sinh các yêu cầu độc lập Client và các khóa riêng
- Nhận các yêu cầu PKCS #10 định dạng PEM từ các Server
- Tập hợp các chứng chỉ
- Hỗ trợ 2 phương thức khác nhau cho thu hồi chứng chỉ
- Tìm kiếm chứng chỉ
- Kiểm tra các chứng chỉ người dùng trên trình duyệt.

1.3.4 Vòng đời của các đối tượng

OpenCA hoạt động an toàn dựa trên sự an toàn của khóa riêng. Có nghĩa là các đối tượng có liên quan đến từng cặp khóa riêng/công khai cần được kết nối với nhau.

Nếu một chứng chỉ đơn sai thì không có vấn đề gì cả, nhưng nếu một khóa được thỏa hiệp thì tất cả các yêu cầu và chứng chỉ liên quan đến khóa đó sẽ bị ảnh hưởng.



Hình 1.3.4.1 Vòng đời của các đối tượng

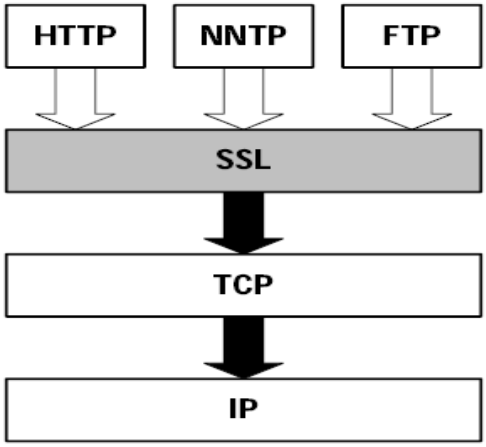
1.4 Công nghệ SSL

1.4.1 Giới thiệu về SSL

SSL là giao thức đa mục đích, được thiết kế để tạo ra các giao tiếp giữa hai chương trình ứng dụng trên một cổng định trước (Socket 443), nhằm mã hoá toàn bộ thông tin gửi / nhận. Giao thức SSL được hình thành và phát triển đầu tiên năm 1994 bởi nhóm nghiên cứu Netscape, dẫn dắt bởi Elgamal và nay đã trở thành chuẩn bảo mật cài đặt trên Internet.

SSL được thiết kế độc lập với tầng ứng dụng, để đảm bảo tính bí mật, an toàn và chống giả mạo luồng thông tin qua Internet giữa hai ứng dụng bất kỳ, thí dụ giữa Webserver và các trình duyệt (Browsers), do đó được sử dụng rộng rãi trong nhiều ứng dụng khác nhau trên môi trường Internet.

Toàn bộ cơ chế và hệ thống thuật toán mã hoá trong SSL được phổ biến công khai, trừ khoá phiên (Session key) được sinh ra tại thời điểm trao đổi giữa hai ứng dụng là ngẫu nhiên và bí mật đối với người quan sát trên mạng máy tính. Ngoài ra, giao thức SSL còn đòi hỏi người dùng phải được chứng thực bởi đối tượng thứ ba (CA) thông qua chứng chỉ số (Digital Certificate) dựa trên mật mã công khai (ví dụ RSA).



Hình 1.4.1.1 Vị trí SSL trong mô hình OSI

SSL được thiết kế như là một giao thức riêng cho vấn đề bảo mật, có thể hỗ trợ cho nhiều ứng dụng. Giao thức SSL hoạt động bên trên TCP / IP và bên dưới các ứng dụng tầng cao hơn như là HTTP (HyperText Transfer Protocol), LDAP (Lightweight Directory Access Protocol) hoặc IMAP (Internet Messaging Access Protocol). Hiện nay SSL được sử dụng chủ yếu cho các giao dịch trên Web.

SSL cho phép một Server (có hỗ trợ SSL) tự xác thực với một Client (cũng hỗ trợ SSL), ngược lại cho phép Client tự xác thực với Server. SSL cho phép cả hai máy thiết lập một kết nối được mã hoá.

- Chứng thực SSL Server: cho phép Client xác thực được Server muốn kết nối. Trình duyệt sử dụng kỹ thuật mã hóa công khai để chắc chắn rằng chứng chỉ và public ID của Server là có giá trị, được cấp phát bởi một CA (trong danh sách các CA tin cậy của Client).
- Chứng thực SSL Client: cho phép Server xác thực được Client muốn kết nối. Server cũng sử dụng kỹ thuật mã hoá khoá công khai để kiểm tra chứng chỉ của Client và public ID là đúng, được cấp phát bởi một CA (trong danh sách các CA tin cậy của Server).

- Mã hoá kết nối: tất cả các thông tin trao đổi giữa Client và Server được mã hoá trên đường truyền, nhằm nâng cao khả năng bảo mật. Điều này rất quan trọng đối với cả hai bên, khi có các giao dịch mang tính riêng tư. Ngoài ra, tất cả các dữ liệu được gửi đi trên một kết nối SSL đã được mã hoá, còn được bảo vệ nhờ cơ chế tự động phát hiện các xáo trộn, thay đổi trong dữ liệu.

Giao thức SSL gồm hai tầng:

- Tầng thấp nhất là tầng SSL Record Protocol. Nó được sử dụng để đóng gói một số giao thức ở mức cao hơn. Một trong những giao thức được đóng gói là SSL
- Tầng thứ 2 là tầng Handshake Protocol. Nó là giao thức cho phép Server và Client xác thực lẫn nhau. Chúng thoả thuận thuật toán mã hoá và các khoá mật mã trước khi thực hiện gửi hoặc nhận dữ liệu

1.4.2 Các phiên bản

SSLv2: Phiên bản đầu tiên của giao thức SSL do Netscape Corporation thiết kế.

SSLv3: Phiên bản SSL version 3.0 do Netscape Corporation thiết kế, đã có trợ giúp Chain certificate (chứng chỉ nhóm), được hỗ trợ cho các trình duyệt phổ thông.

TLSv1: Giao thức Transport Layer Security version 1.0 dựa trên cơ sở của SSLv3, thiết kế bởi IETF, nhưng hiện chưa được hỗ trợ cho tất cả các trình duyệt thông dụng.

1.4.3 Các thuộc tính cơ bản

Kết nối an toàn:

Quá trình mã hóa dữ liệu được áp dụng sau khi quá trình bắt tay (Handshake) đầu tiên xác định được khoá bí mật. Mật mã đối xứng được sử dụng cho quá trình mã hoá dữ liệu (DES, RC4…). Đảm bảo thông tin không thể bị truy cập bởi đối tượng thứ ba.

Danh tính của người bên kia có thể được xác thực bằng mật mã khoá công khai (RSA, DSS…).

Kết nối tin cậy:

Vận chuyển thông điệp bao gồm quá trình kiểm tra tính toàn vẹn của thông điệp sử dụng hàm kiểm tra MAC có khoá. Các hàm băm an toàn (ví dụ SHA, MD5…)

được dùng cho quá trình thực hiện hàm MAC, nhằm đảm bảo thông tin không bị sai lệch và thể hiện chính xác thông tin gốc gửi đến.

1.4.4 Mục đích

Khả năng an toàn: SSL được sử dụng để thiết lập kết nối an toàn giữa hai nhóm.

Khả năng tương tác giữa các phần tử: Các nhà lập trình độc lập có thể phát triển các ứng dụng sử dụng SSL 3.0, sau khi trao đổi các tham số mật mã mà không phải biết mã chương trình của các ứng dụng khác.

Khả năng mở rộng: SSL cung cấp một framework, trong đó các phương pháp mã hoá và mã hoá khóa công khai kết hợp chặt chẽ với nhau.

1.4.5 Bảo mật của SSL

Mức độ bảo mật của SSL phụ thuộc chính vào độ dài khoá hay phụ thuộc vào việc sử dụng phiên bản mã hoá 40bit và 128bit. Phương pháp mã hoá 40bit được sử dụng rộng rãi không hạn chế ngoài nước Mỹ, phiên bản mã hoá 128bit chỉ được sử dụng trong nước Mỹ và Canada. Theo luật pháp Mỹ, các mật mã "mạnh" được phân loại vào nhóm "vũ khí" (weapon) và do đó khi sử dụng ngoài Mỹ (coi như là xuất khẩu vũ khí) phải được phép của chính phủ Mỹ hay phải được cấp giấy phép của Bộ Quốc phòng Mỹ (DoD).

Đây là lợi điểm cho quá trình thực hiện các dịch vụ thương mại và thanh toán điện tử trong Mỹ và các nước đồng minh phương Tây, là điểm bất lợi cho việc sử dụng các sản phẩm cần có cơ chế bảo mật và an toàn trong giao dịch điện tử nói chung và thương mại điện tử nói riêng trong các nước khác.

Các phương thức tấn công (hay bẻ khoá) nhằm vào các thuật toán bảo mật thường dựa trên phương pháp "tấn công vét cạn" (Brute-force attack) bằng cách thử-sai miền không gian các giá trị có thể của khoá. Số phép thử-sai tăng lên khi độ dài khoá tăng và dẫn đến vượt quá khả năng và công suất tính toán, kể cả các siêu máy tính hiện đại. Thí dụ, với độ dài khoá là 40bit, thì số phép thử sẽ là 2^{40}=1,099,511,627,776 tổ hợp.

Tuy nhiên độ dài khoá lớn kéo theo tốc độ tính toán giảm (luỹ thừa nghịch đảo) và dẫn đến khó có khả năng áp dụng trong thực tiễn. Một khi khoá bị phá, toàn bộ thông tin giao dịch trên mạng sẽ bị kiểm soát.

Tuy nhiên do độ dài khoá lớn (thí dụ 128, 256 bít), số phép thử-sai trở nên "không thể thực hiện" vì phải mất hàng năm hoặc thậm chí hàng nghìn năm với công suất và năng lực tính toán của máy tính mạnh nhất hiện nay.

Ngay từ năm 1995, bản mã hoá 40bit đã bị phá bởi sử dụng thuật toán vét cạn. Ngoài ra, một số thuật toán bảo mật (DES 56bit, RC4, MD4,...) hiện nay cũng bị coi là không an toàn khi áp dụng một số phương pháp và thuật toán tấn công đặc biệt. Đã có một số đề nghị thay đổi trong luật pháp Mỹ, nhằm cho phép sử dụng rộng rãi các phần mềm mã hoá dùng mã 56bit, nhưng hiện nay vẫn chưa được chấp thuận.

1.4.6 Ưu điểm và hạn chế của SSL

❖ **Ưu điểm của SSL**

Tính năng mạnh nhất của SSL / TLS là chúng xác định mối quan hệ với các tầng giao thức khác như thế nào trong hệ thống kiến trúc mạng OSI. Tại mức cao nhất là phần mềm ứng dụng hoặc các trình duyệt. Chạy phía dưới các ứng dụng này là giao thức tầng ứng dụng bao gồm Telnet, FTP, HTTP…

Bên dưới nữa là giao thức SSL và các thuật toán mã hoá được sử dụng để kết nối. Bên dưới SSL là tầng giao vận. Hầu hết các trường hợp đó là TCP/IP. Tuy nhiên, giao thức SSL là duy nhất, không phụ thuộc vào giao thức mạng. Bởi vì SSL không phụ thuộc vào các tầng giao thức, cho nên SSL trở thành một nền tảng độc lập hay là một thực thể mạng độc lập.

Một sức mạnh khác của SSL là ngăn chặn cách thức tấn công từ điển. Cách thức này sử dụng từ điển để phá khoá trong hệ mã hoá. SSL khắc phục được điều này bởi cho phép không gian khoá là rất lớn đối với hệ mã hoá được sử dụng. SSL cung cấp hai mức độ tin cậy: 40 bit và 128 bit tuỳ thuộc khả năng của browser. SSL 128 bit và SSL 40 bit ý nói độ dài của khoá phiên dùng để mã hoá dữ liệu sau khi đã định danh và được thiết lập bằng giải thuật khoá công khai (RSA hoặc Diffie-Hellman). Độ dài của khoá phiên càng lớn thì độ bảo mật càng cao. Hiện nay SSL 128 bit có độ tin

cậy lớn nhất. Theo RSA phải mất hàng tỉ năm mới có thể giải mã được bằng các kỹ thuật hiện nay.

Cách thức tấn công "từ điển" có thể bị ngăn chặn bởi sử dụng phương pháp số Nonce (Nonce number). Số này được sinh ngẫu nhiên, được server sử dụng, Nonce number là một số không thể bị phá khoá.

Giao thức SSL còn bảo vệ chính nó với đối tác thứ 3. Đó là các client xâm nhập bất hợp pháp dữ liệu trên đường truyền. Client xâm nhập này có thể giả mạo client hoặc server, SSL ngăn chặn sự giả mạo này bằng cách sử dụng khoá riêng của server và sử dụng chứng chỉ số. Phương thức bắt tay trong TLS cũng tương tự. Tuy nhiên, TLS tăng cường sự bảo mật bằng cách cho phép truyền phiên bản giao thức, số hiệu phiên làm việc, hệ mã hoá và cách thức nén được sử dụng. TLS bổ xung thêm hai thuật toán băm không có trong SSL.

❖ 2. Hạn chế của SSL

Giao thức SSL, cũng giống như bất kỳ công nghệ nào, cũng có những hạn chế. Vì SSL cung cấp các dịch vụ bảo mật, cần quan tâm đặc biệt tới các giới hạn của nó. Giới hạn của SSL thường là trong ba trường hợp:

Do những ràng buộc cơ bản của bản thân giao thức SSL. Đây là hệ quả của việc thiết kế SSL và ứng dụng chịu tác động của nó.

Do giao thức SSL cũng thừa kế một vài điểm yếu từ các công cụ mà nó sử dụng, cụ thể là các thuật toán ký và mã hoá. Nếu các thuật toán này có điểm yếu, SSL thường không thể khắc phục chúng.

Do môi trường, trong đó SSL được triển khai, có những thiếu sót và giới hạn.

Mặc dù trong thiết kế đã xét đến mối quan hệ với rất nhiều ứng dụng khác nhau, SSL rõ ràng được tập trung vào việc bảo mật các giao dịch Web. SSL yêu cầu một giao thức vận chuyển tin cậy như TCP. Đó là yêu cầu hoàn toàn hợp lý trong các giao dịch Web, vì bản thân HTTP cũng yêu cầu TCP. Tuy nhiên, điều này cũng có nghĩa là SSL không thể thực thi mà sử dụng giao thức vận chuyển không kết nối như UDP. Vì vậy, SSL có thể hoạt động hiệu quả với phần lớn các ứng dụng thông thường. Và thực tế hiện nay, SSL đang được sử dụng cho nhiều các ứng dụng bảo mật, bao gồm truyền file, đọc tin mạng, điều khiển truy cập từ xa...

Một đặc điểm khác khiến SSL bị lỗi khi hỗ trợ dịch vụ bảo mật đặc biệt như là Non-repudiation (không chối bỏ). Dịch vụ này ngăn ngừa bên tạo dữ liệu và bên ký dữ liệu từ chối hay phủ nhận điều mình đã thực hiện. SSL không cung cấp các dịch vụ non-repudiation, do đó sẽ không phù hợp với các ứng dụng yêu cầu dịch vụ này.

Điểm yếu của mã hoá SSL còn do phiên làm việc tồn tại quá lâu trong quá trình bắt tay, khoá phiên được khởi tạo giữa client và server được dùng trong suốt quá trình kết nối. Khi khoá này còn tồn tại, mỗi khi thông điệp được gửi, tồn tại một lỗ hổng bảo mật trong kết nối cho phép xâm nhập. Giao thức TLS khắc phục được lỗi này bằng cách thay đổi khoá cho mỗi phiên làm việc.

Các giao tiếp thực giữa client và server cũng là mục tiêu tấn công vì chúng lưu trữ các thông điệp giữa hai điểm đầu cuối. Thông điệp trong SSL được mã hoá, tuy nhiên tại mỗi điểm đầu cuối thông điệp được giải mã, SSL chưa có cơ chế duy trì sự mã hoá trong bộ nhớ đệm của hệ thống tương ứng.

Vấn đề khác của SSL là khả năng áp dụng đối với người sử dụng trên toàn cầu. Mặc dù một vài client trên các nước khác có hỗ trợ kiến trúc SSL, nhưng vẫn còn hạn chế về ranh giới của sự mã hoá. Ranh giới này do chính phủ Mỹ đưa ra, nó giới hạn số lượng bit được sử dụng trong các hệ mã hoá. SSL có hỗ trợ mã hoá 128 bit trong các phiên giao dịch toàn cầu, nhưng thực tế chỉ sử dụng hệ mã hoá 40 bit. Các hạn chế về bảo mật này càng cho phép kẻ tấn công có nhiều cơ hội hơn khi tìm cách bẻ khoá hệ thống.

Một vài ý kiến lại cho rằng hạn chế lớn nhất của SSL không chỉ ở giao thức bắt tay, mà tồn tại trong tầng bản ghi của giao thức. Trong quá trình bắt tay, việc chứng thực giữa client và server thực hiện rất nghiêm ngặt, do dùng chứng chỉ số và khoá. Tuy nhiên, trong tầng bản ghi, quá trình xác thực không được thực hiện trong suốt giai đoạn kết nối còn lại. Do không có sự xác thực giữa client và server, dẫn đến kẻ tấn công có thể mạo danh client hoặc server trong quá trình kết nối.

Nhiều ý kiến cho rằng SSL chỉ giới hạn cho các ứng dụng thương mại điện tử, điều này là không đúng. Vì các tổ chức tài chính có thể sử dụng SSL để truyền số PIN, các công ty bảo hiểm sử dụng SSL để truyền dữ liệu khách hàng, các công ty hoạt động theo mô hình B2B (Bussiness-to-Bussiness) sử dụng SSL xây dựng các

phiên giao dịch giữa các công ty, SSL có thể được sử dụng trong một tổ chức để truyền dữ liệu trên mạng cục bộ.

Do hạn chế về công nghệ, vẫn còn một số server không thể dùng mã hoá SSL. Mặc dù ý tưởng mã hoá là quan trọng, tuy nhiên có sự hạn chế trong sức mạnh của server trong quá trình kiểm tra chữ ký số và thực hiện ký số. Do không đáp ứng được yêu cầu xử lý, nhiều web server gặp khó khăn trong thực hiện kết nối SSL. Điều này khó có thể chấp nhận được đối với người sử dụng và khách hàng.

Các chứng chỉ server tự ký có thể cung cấp bảo mật, nhưng không xác thực. Một chứng chỉ tự ký không được chứng nhận bởi máy người dùng và không qua các bước thêm vào sự tin cậy cho chứng chỉ server bằng tay. Theo mặc định, các máy tính Windows tin tưởng các chứng chỉ server chỉ khi từ các CA chỉ định như là VeriSign.

Về phía client, thiết lập mặc định cho các browser phổ biến như Internet Explorer và Nescape không kiểm tra sự thu hồi chứng chỉ và vẫn chấp nhận các phiên SSL 2.0. Thêm vào đó, các thiết lập mặc định thường cho phép các trang mã hoá SSL được lưu trong browser cache mà không cần mã hoá.

Chương 2: CHỮ KÝ SỐ VÀ CHỨNG CHỈ SỐ

2.1 Khái niệm chữ ký số

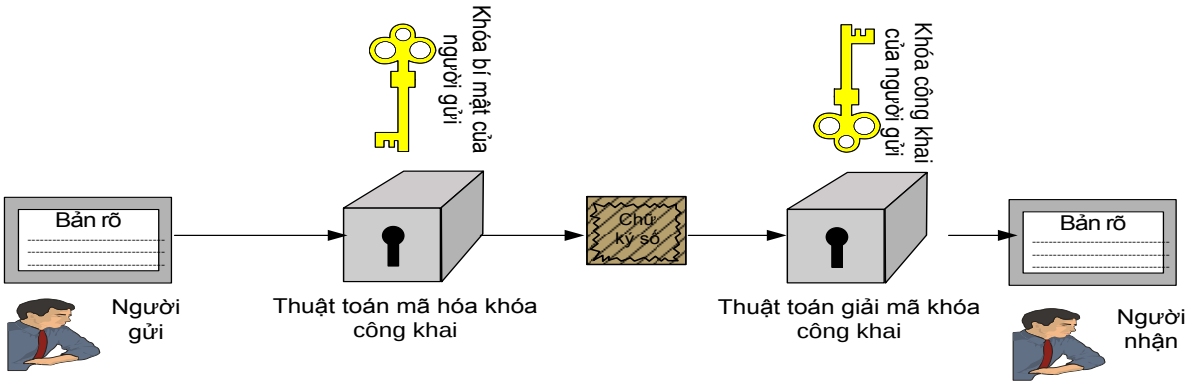
Với thỏa thuận thông thường trên giấy, hai đối tác xác nhận sự đồng ý bằng cách ký tay vào cuối các hợp đồng. Bằng cách nào đó người ta phải thể hiện đó là chữ ký của riêng họ và kẻ khác không thể giả mạo. Mọi cách sao chép chữ ký trên giấy thường dễ bị phát hiện, vì bản sao có thể phân biệt được với bản gốc.

Các giao dịch hợp tác trên mạng cũng được thực hiện theo cách tương tự, nghĩa là hai đối tác trên hai nút mạng cũng phải ký vào Bản thỏa thuận. Chỉ khác là văn bản truyền trên mạng được biểu diễn dưới dạng "số" (chỉ dùng chữ số 0 và 1), ta gọi nó này là "văn bản số" (điện tử). Do đó chữ ký trên "văn bản số" khác với chữ ký trên văn bản giấy thông thường.

Việc giả mạo và sao chép lại đối với "văn bản số" là việc hoàn toàn dễ dàng, không thể phân biệt được bản gốc với bản sao. Như vậy "chữ ký" ở cuối "văn bản số" không thể chịu trách nhiệm đối với toàn bộ nội dung văn bản loại này. Do đó Chữ ký thể hiện trách nhiệm đối với toàn bộ "văn bản số" phải là "chữ ký" được ký trên từng bit của văn bản loại này. Bản sao của "chữ ký số" có tư cách pháp lí.

Chữ ký thông thường được kiểm tra bằng cách so sánh nó với chữ ký gốc. Ví dụ, ai đó ký một tấm séc để mua hàng, người bán phải so sánh chữ ký trên mảnh giấy với chữ ký gốc nằm ở mặt sau của thẻ tín dụng để kiểm tra. Dĩ nhiên, đây không phải là phương pháp an toàn vì nó dễ dàng bị giả mạo.

"Chữ ký số" có thể được kiểm tra chính xác nhờ dùng một thuật toán kiểm tra công khai. Như vậy, bất kỳ ai cũng có thể kiểm tra được chữ ký số. Việc dùng một sơ đồ chữ ký an toàn có thể sẽ ngăn chặn được khả năng giả mạo.



Hình 2.1.1 Mô hình chữ ký số

2.2 Đại diện thông điệp

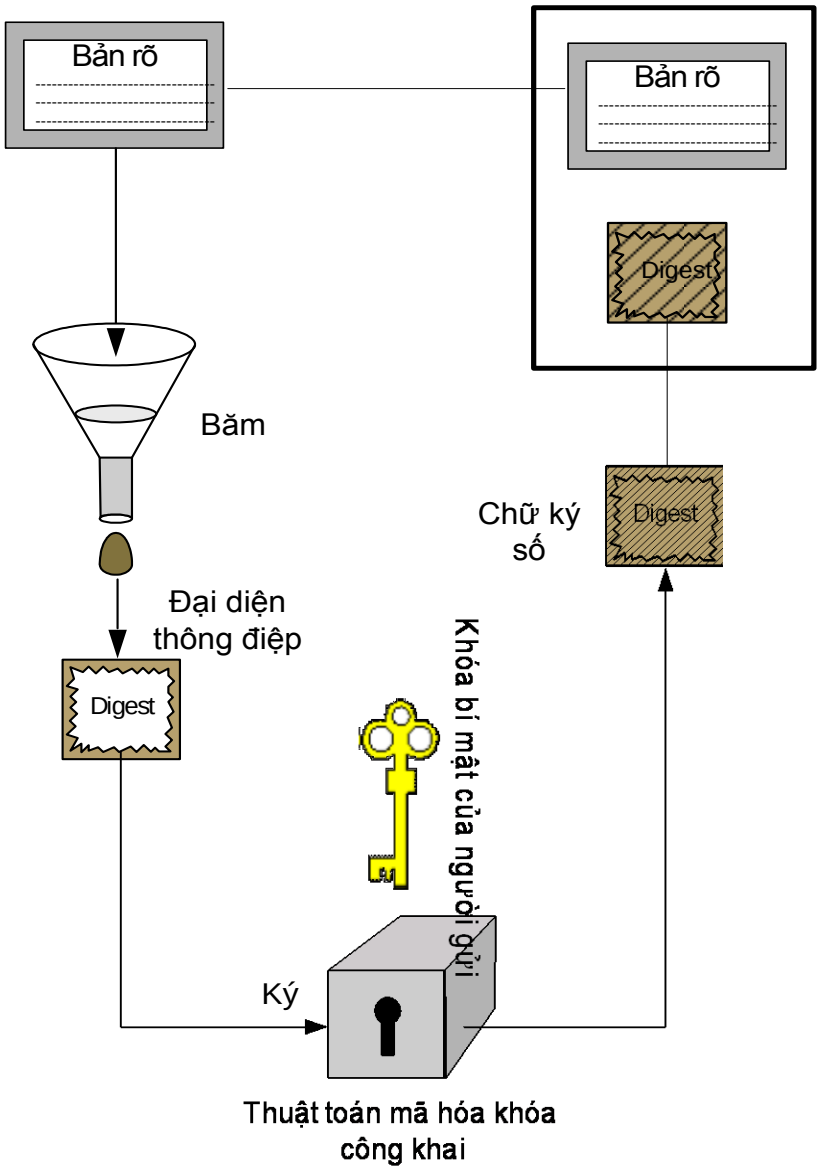
Vì "Chữ ký số " được ký trên từng bit của "văn bản số", nên độ dài của nó ít nhất cũng bằng văn bản cần ký. Như vậy sẽ tốn kém bộ nhớ cũng như thời gian "ký" và thời gian truyền "Chữ ký số ". Trên thực tế thay vì ký trên "văn bản số", người ta ký trên "Đại diện" (Digest) của nó.

Để ký trên "văn bản số" dài, đầu tiên phải tạo "đại diện" của văn bản nhờ "Hàm băm ". Một thông điệp được đưa qua hàm băm sẽ tạo ra xâu bit với độ dài cố định và ngắn hơn được gọi là "Đại diện" (Digest). Mỗi thông điệp đi qua 1 hàm băm chỉ cho duy nhất 1 "Đại diện". Ngược lại, "khó" tìm được 2 thông điệp khác nhau mà có cùng một "Đại diện" (ứng với cùng 1 hàm băm).

Hàm băm kết hợp với "chữ ký số" ở trên sẽ tạo ra một loại "chữ ký điện tử " vừa an toàn (không thể cắt / dán), vừa có thể dùng để kiểm tra tính toàn vẹn của thông điệp.

1). Người gửi: Tạo ra "chữ ký số".

- Đưa thông điệp cần gửi qua hàm băm tạo ra "Đại diện".
- Mã hoá "Đại diện" bằng khoá riêng (private) của người gửi để tạo ra "chữ ký số".
- Mã hoá thông điệp và chữ ký bằng khoá công khai (public) của người nhận, gửi đi.

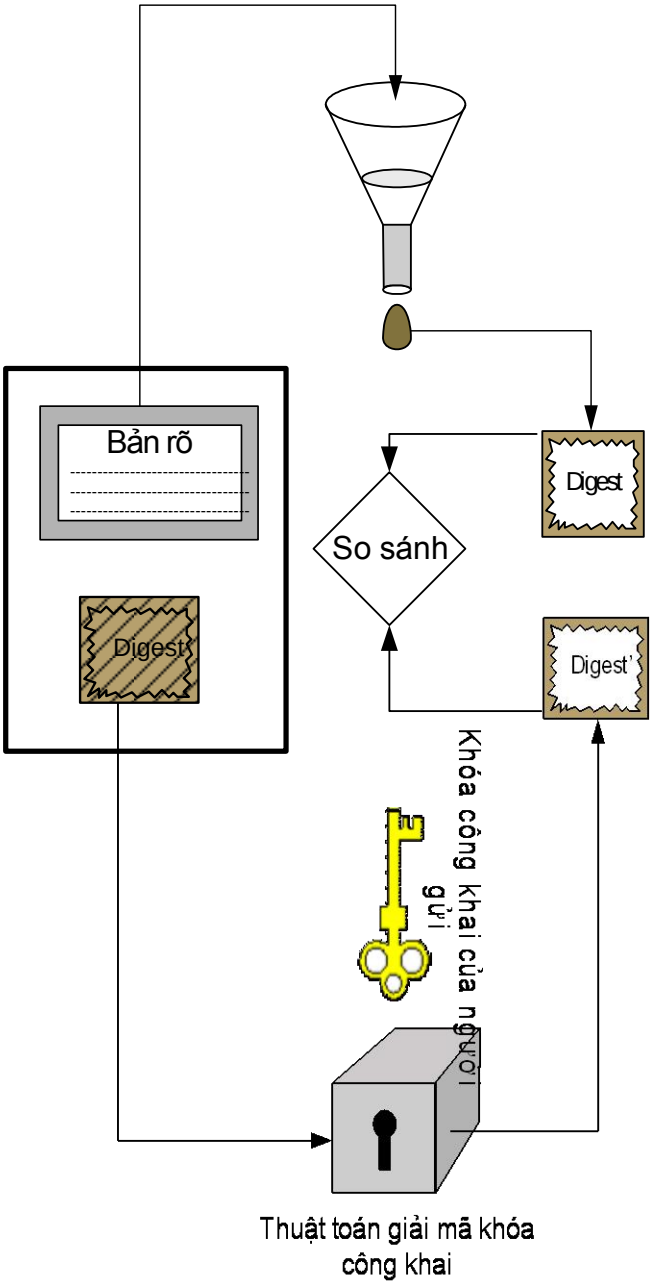


Hình 2.2.1 Mô hình mã hóa thông điệp và chữ ký bằng khóa bí mật

2). Người nhận: Định danh người ký, kiểm tra tính toàn vẹn của thông điệp.

- Giải mã thông điệp bằng khoá riêng của mình, giải mã chữ ký bằng khoá công khai của người gửi để lấy "Đại diện" ra.
- Cho thông điệp qua hàm băm để tạo ra "Đại diện" mới.
- So sánh "Đại diện" mới với "Đại diện" nhận được.

Nếu chúng giống nhau thì người nhận có thể vừa định danh được người gửi, vừa kiểm tra tính toàn vẹn của thông điệp.



Hình 2.2.2 Mô hình giải mã thông điệp và chữ ký bằng khóa công khai

2.3 Khái niệm chứng chỉ số

Việc sử dụng mã hóa hay ký số chỉ giải quyết được vấn đề bảo mật và xác thực thông điệp. Tuy nhiên "khó" thể đảm bảo rằng người ký là đối tác thật. Trong nhiều trường hợp cần thiết phải "chứng minh" bằng phương tiện điện tử danh tính của ai đó. Ví dụ phải "chứng minh" rằng người người ký là "chủ đích thực" hiện thời của chìa khóa ký.

Một cách giải quyết là dùng "Chứng chỉ số" để xác nhận "chủ đích thực" hiện thời của khóa công khai.

Chứng chỉ số là một tệp tin điện tử dùng để nhận diện một cá nhân, một máy dịch vụ, một thực thể nào đó. Nó gắn định danh của đối tượng đó với một khóa công khai, giống như bằng lái xe, hộ chiếu, chứng minh thư.

Chứng chỉ số là kết quả của dự án phát triển chuẩn thư mục X.500 của ITU-T phát triển vào cuối những năm thập niên 90. Chứng chỉ số được ITU-T đặc tả trong tài liệu X.509 và dần được thay đổi qua các phiên bản cho phù hợp với thực tế. Hiện nay Chứng chỉ X.509 phiên bản 3 được sử dụng trong các hệ thống xác thực.

Một nơi có thể chứng nhận các thông tin của một thực thể là đúng, nó được gọi là cơ quan xác thực chứng chỉ (Certificate Authority - CA). Đó là một đơn vị có thẩm quyền xác nhận định danh và cấp các chứng chỉ số. CA có thể là một đối tác thứ ba độc lập hoặc tổ chức tự vận hành một hệ thống tự cấp các chứng chỉ cho nội bộ.

Các phương pháp để xác định định danh phụ thuộc vào các chính sách mà CA đặt ra. Chính sách lập ra phải đảm bảo việc cấp chứng chỉ số phải đúng đắn, ai được cấp và mục đích dùng vào việc gì. Thông thường, trước khi cấp một chứng chỉ số, CA sẽ công bố các thủ tục cần thiết phải thực hiện cho các loại chứng chỉ số.

Chứng chỉ số chứa khóa công khai, được gắn với một tên duy nhất của một đối tượng (như tên của một cá nhân hay máy dịch vụ). Chứng chỉ số giúp ngăn chặn việc sử dụng khóa công khai cho việc giả mạo. Chỉ có khóa công khai được chứng thực bởi chứng chỉ số sẽ làm việc với khóa bí mật tương ứng. Nó được sở hữu bởi đối tượng với định danh đã được ghi trong chứng chỉ số.

Ngoài khóa công khai, chứng chỉ số còn chứa thông tin về đối tượng như tên mà nó nhận diện, hạn dùng, tên của CA cấp chứng chỉ số, mã số … Quan trọng nhất là chứng chỉ số phải có "chữ ký số" của CA đã cấp chứng chỉ đó. Giống như chứng chỉ đã được "đóng dấu", để cho người dùng khóa công khai có thể kiểm tra.

Một người muốn sử dụng Hệ mã hóa khóa công khai để mã hóa thông báo và gửi cho người nhận, người gửi phải có bản sao khóa công khai của người nhận.

Một người muốn kiểm tra chữ ký số của người khác, họ phải có bản sao khóa công khai của người ký.

Chúng ta gọi cả hai thành viên (mã hóa thông báo và kiểm tra chữ ký số) là những người sử dụng khóa công khai.

Khi khóa công khai được gửi đến người sử dụng khóa công khai, thì không cần thiết phải giữ bí mật khóa công khai này. Tuy nhiên, người sử dụng khóa công khai phải đảm bảo rằng khóa công khai được sử dụng đúng là của đối tác. Nếu kẻ phá hoại dùng khóa công khai khác thay thế cho khóa công khai hợp lệ, thì nội dung thông báo đã mã hóa có thể bị lộ, chữ ký số có thể bị làm giả. Rõ ràng khóa công khai cần phải được xác thực trước khi dùng.

Đối với nhóm thành viên nhỏ, yêu cầu trên có thể được thỏa mãn dễ dàng. Ví dụ trường hợp hai người quen biết nhau, khi người này muốn truyền thông an toàn với người kia, họ có thể có được bản sao khóa công khai của nhau bằng cách trao đổi các đĩa nhớ có ghi các khóa công khai của từng người. Như vậy đảm bảo rằng các khóa công khai được lưu giữ an toàn trên mỗi hệ thống cục bộ của từng người. Đây chính là hình thức phân phối khóa công khai thủ công.

Phân phối khóa công khai thủ công như trên là không thực tế hoặc không thỏa đáng khi số lượng người dùng là quá lớn và nơi làm việc phân tán. Hệ thống cấp chứng chỉ khóa công khai giúp cho việc phân phối khóa công khai có hệ thống và chuẩn mực.

2.4 Hệ thông cung cấp chứng chỉ khóa công khai

CA phát hành các chứng chỉ cho những người nắm giữ cặp khóa công khai và khóa riêng. Chứng chỉ gồm có khóa công khai và thông tin dùng để nhận dạng duy nhất chủ thể (subject) của chứng chỉ. Chủ thể của chứng chỉ có thể là một người, thiết bị, hoặc một thực thể có nắm giữ khóa riêng tương ứng. Khi chủ thể của chứng chỉ là một người hoặc một thực thể nào đó, chủ thể thường được nhắc đến như là một thực thể (subscriber) của CA. Các chứng chỉ được CA ký, bằng khóa riêng của CA.

Một khi các chứng chỉ số được thiết lập, công việc của người sử dụng khóa công khai rất đơn giản. Giả thiết rằng, họ đã có khóa công khai của CA (ví dụ: thông qua phân phối khóa công khai thủ công) và tin cậy CA phát hành các chứng chỉ hợp lệ. Nếu người dùng cần khóa công khai của một thuê bao nào đó của CA, anh ta có thể thu được khóa công khai của thuê bao bằng cách tìm trong bản sao chứng chỉ của họ,

lấy ra khóa công khai. Tất nhiên trước đó anh ta phải kiểm tra chữ ký trên chứng chỉ có đúng là của CA không.

Hệ thống cấp chứng chỉ như trên là đơn giản và kinh tế khi được thiết lập trên diện rộng và tự động, bởi vì một trong các đặc tính quan trọng của chứng chỉ là:

"Các chứng chỉ có thể được phát hành mà không cần phải bảo vệ thông qua các dịch vụ an toàn truyền thông để đảm bảo xác thực và toàn vẹn".

Chúng ta không cần giữ bí mật khóa công khai, như vậy chứng chỉ không phải là bí mật. Hơn nữa, ở đây không đòi hỏi các yêu cầu về tính xác thực và toàn vẹn do các chứng chỉ tự bảo vệ. Chữ ký của CA trong chứng chỉ đã cung cấp tính xác thực và toàn vẹn. Người dùng khóa công khai trong các chứng chỉ như trên được gọi là thành viên tin cậy.

Kẻ truy nhập trái phép định làm giả chứng chỉ khi chứng chỉ này đang lưu hành cho những người sử dụng khóa công khai, họ sẽ phát hiện ra việc làm giả, bởi vì chữ ký của CA có thể được kiểm tra chính xác. Chính vì thế các chứng chỉ khóa công khai được phát hành theo cách không an toàn, ví dụ như thông qua các máy chủ, các hệ thống thư mục, các giao thức truyền thông không an toàn.

Lợi ích cơ bản của hệ thống cấp chứng chỉ là: người sử dụng khóa công khai có được số lượng lớn các khóa công khai của nhiều người dùng một cách tin cậy, nhờ khóa công khai của CA. Lưu ý rằng chứng chỉ số chỉ có nghĩa khi CA phát hành các chứng chỉ hợp lệ.

Chương 3: CA (CERTIFICATE AUTHORITY)

3.1 Giới thiệu một số vấn đề liên quan đến cơ sở hạ tầng khóa công khai

3.1.1 Các giao thức quản ý cơ sở hạ tầng khóa công khai theo chuẩn X509

PKI được xây dựng bao gồm rất nhiều mô hình riêng biệt và việc quản trị các trong các mô hình đó là khác nhau. Management protocol được đưa ra bởi nó cần thiết để hỗ trợ các tương tác online giữa các thành phần PKI (giữa CA và hệ thông client, giữa các CA phát hành cross-certificates).

Trước khi xác định rõ riêng biệt các định dạng message và các thủ tục cho phần mềm PKI chúng ta phải đi xây dựng mô hình PKI Management: định nghĩa các thực thể trong PKI Management và tương tác của chúng. Sau đó chúng ta đi nhóm các tính năng này cho phù hợp các kiểu có thể định danh của các thực thể đầu cuối (end entity).

Các thực thể được đưa ra trong PKI Management bao gồm end entities (ví dụ, thực thể được đặt tên trong trường Subject của certificates) và CA (ví dụ, thực thể được đặt tên trong trường Issuer của certificates). Dưới đây là một vài ví dụ về các định nghĩa trong PKI Management.

Subjects và End Entities

Như đã đề cập ở trên thì thuật ngữ "subject" được sử dụng ở đây để tham chiếu tới một thực thể được đặt tên trong trường Subject của một certificate, khi chúng ta muốn phân biệt giữa các công cụ hay giữa các phần mềm được sử dụng bởi subject đó (ví dụ, một moduel quản lý certificate cục bộ) được gọi là ""subject equipment". Trong trường hợp tổng quát chúng ta sử dụng thuật ngữ "End Entity" (EE).

Tất cả các EEs yêu cầu bảo mật cục bộ truy cập tới một số thông tin tối thiểu: tên sở hữu và private key, tên của CA được tin cậy bởi thực thể và public key của CA (hoặc fingerprint của public key). Nơi lưu trữ các thông tin này có thể thay đổi, sự thay đổi này tùy thuộc vào cách cài đặt và ứng dụng (ví dụ, dạng file như crytographic tokens), nơi này được gọi là môi trường an toàn cá nhân (PSE) của EE, định dạng của PSE nằm ngoài phạm vi của RFC này.

Certificate Authority

Certificate Authority (CA) là một "third party" thực sự hoặc cũng có thể không phải là "third party" (điều này cho phép chúng ta phân biệt RootCA và Non-RootCA),

CA thường thuộc về một tổ chức nào đó nhằm mục đích hỗ trợ các EEs. Một lần nữa chúng ta sử dụng thuật ngữ CA để chỉ thực thể được đặt tên trong trường Issuer của certificate, khi cần phân biệt các công cụ phần cứng hoặc phần mềm sử dụng bởi CA chúng ta đưa ra thuật ngữ "CA equipment". CA equipment bao gồm cả 2 thành phần: online và offline (private key của CA được gọi là thành phần offline).

Các yêu cầu về PKI Management

Bao gồm 13 yêu cầu sau đây :

- Tương thích với chuẩn ISO 9594-8 và các phần certificate extensions.
- Tương thích giữa các thành phần trong các series.
- Đơn giản trong vấn đề cập nhật key pair mà không ảnh hưởng đến key pair khác (trong hệ thống).
- Sử dụng tính tin cậy trong PKI Management protocols phải dễ dàng các bài toán điều tiết.
- Phải tương thích với các thuật toán mã hóa (chuẩn công nghiệp) : RSA, DSA, SHA-1, ...
- Không loại trừ việc sinh cặp khóa bởi EEs, RAs, CAs.
- Hỗ trợ việc công khai các certificates (tùy thuộc vào cài đặt khác nhau và các môi trường khác nhau).
- Hỗ trợ việc hủy bỏ certificate của EEs (CRLs).
- Có thể sử dụng đa dạng ‘‘transport mechanisms’’: mail, http, TCP/IP và ftp.
- Chỉ có CA mới có thể thay đổi hoặc thêm giá trị trường trong certificate, xóa hoặc thay đổi extension dựa trên các chính sách hoạt động của nó.
- Hỗ trợ công việc cập nhật CA key cho các EEs.
- Các chức năng của RA phụ thuộc vào CA của nó (các cách cài đặt và các môi trường khác nhau).
- Khi EE yêu cầu một certificate bao gồm có cả giá trị public key, thì phải có một giá trị private key tương ứng (ký lên request – Proof of Possession of Private Key).

3.1.2 Hồ sơ chứng chỉ số và CRL(Danh sách hủy bỏ chứng chỉ) cho cơ sở hạ tầng khóa công khai theo chuẩn X509

X509 v3 certificate

Như đã biết, user có một public key sẽ có một private key được sở hữu bởi đúng subject (người dùng hoặc hệ thống) với một kỹ thuật mã hóa và chữ ký số được sử dụng. Tính tin cậy này được sử dụng trong các chứng chỉ public key (gọi là certificate), bị ràng buộc bởi chữ ký của CA (trusted CA) với một khoảng thời gian sử dụng xác định. Certificate có thể được phân phối qua các truyền thông không cần sự tin cậy và các hệ thống server khác nhau và có thể được lưu trong một kho không bảo mật trên hệ thống sử dụng certificate. ANSI X9 đã phát triển định dạng X509 v3 dựa trên việc mở rộng một số trường dự trữ, các trường này bao gồm: thông tin định danh, thông tin về thuộc tính khóa, thông tin về chính sách (policy) hệ thống CA và các bắt buộc certification path (trường basicConstraints).

Certification paths anh trust

Một user của một dịch vụ bảo mật có một public key (có hiệu lực) sẽ có một certificate được chứng nhận bởi một CA (ký tên public key), CA này cũng có thể được chứng nhận bởi một (hoặc nhiều) CA khác. Do vậy, nảy sinh khái niệm về certification path. Trong RFC1422 đã định nghĩa một cấu trúc chuỗi các CAs một cách cứng nhắc, cấu trúc này tương thích với X509 v1, gồm có 3 kiểu CA là: IPRA (Internet Policy Registration Authority), PCAs (Policy Certiification Authorities) và CAs (Certification Authorities). Cấu trúc này có các hạn chế sau: cơ chế top-down tức là tất cả các certification paths phải bắt đầu từ IPRA, quy tắc đặt tên nhánh hạn chế subject của CA, sử dụng khái niệm PCA tức là yêu cầu phải biết từng PCAs được thiết lập trong logic kiểm tra chuỗi certificate. Với X509 v3, thì hầu hết các yêu cầu trên được sử dụng trong certificate extension, mà không cần hạn chế các cấu trúc sử dụng CA. Với cấu trúc này, đưa ra kiến trúc hết sức mềm dẻo cho hệ thống CA.

Revocation

Khi phát hành ra một chứng chỉ, nó đã được định ra một khoảng thời hạn sử dụng nhất định. Tuy nhiên, vì một số lý do nào đó mà người sử dụng muốn hủy bỏ chứng chỉ này khi chưa hết hạn sử dụng. X509 định nghĩa một phương pháp hủy bỏ certificate, phương pháp này cho phép các CAs chấp nhận hủy bỏ chứng chỉ, được gọi

là một CRL (Certificate Revocation List). Danh sách này liệt kê tất cả các chứng chỉ bị hủy bỏ (theo số serial). Khi một hệ thống bảo mật sử dụngchứng chỉ, thì hệ thống này không những kiểm tra chữ ký trên chứng chỉ và tính hiệu lực của nó mà còn kiểm tra sự có mặt của serial này trong CRL đó (tất nhiên là CRL này phải được cập nhật trên toàn bộ hệ thống theo một định kỳ nào đó). Nếu số serial này có trong CRL thì coi như chứng chỉ đó đã bị hủy bỏ. CRL có thể được phân phối qua các truyền thông không bảo mật và các hệ thống server (repository). Một hạn chế của phương pháp CRL, đó là khoảng thời gian phát hành CRL là không liên tục. Có thể giải quyết hạn chế này bằng các phương pháp trực tuyến (online method), phương pháp này có thể áp dụng trong một số môi trường. Tuy nhiên, để sử dụng các phương pháp này sẽ phải đảm nhiệm thêm một số yêu cầu mới về bảo mật mới.

3.2 Cài đặt thiệt lập cấu hình cho máy CA

Hệ thống cung cấp chứng chỉ số MyCA được xây dựng trên hệ điều hành RedHat Linux, gồm hai mô hình:

- Mô hình cấp phát, quản lý và hủy bỏ chứng chỉ, do người sử dụng sinh khóa
- Mô hình cấp phát, quản lý và hủy bỏ chứng chỉ do trung tâm sinh khóa (mô hình sinh khóa tập trung)

3.2.1 Cài đặt

Đối với các máy được thiết lập làm máy CA (Certificate Authority) trước khi thực hiện việc cài đặt cần kiểm tra một số yêu cầu về phần mềm dưới đây:

- Hệ điều hành RedHat Linux 7.2
- Perl phiên bản 5.6.0 hoặc cao hơn
- Apache phiên bản 1.3.12 hoặc cao hơn

Toàn bộ phần mềm MyCA được lưu trên một đĩa CD ROM. Để cài đặt máy CA người thực hiện có thể tiến hành như sau:

- Copy tệp MayCA.tgz từ đĩa CD vào máy cần thiết lập làm máy CA.
- Gỡ nén tệp MayCA.tgz, bởi lệnh

  tar –xvzf /đường dẫn/MayCA.tgz

được thư mục MayCA, trong đó có các thư mục: MyCA, và myca (trong thư mục này có các thư mục con: cgi-ca,htdocs-ca,cgi-print).

- Copy thư mục myca vào thư mục /home
- Copy thư mục MyCA ra ngoài cùng của hệ thống cây thư mục

3.2.2 Thiết lập cấu hình

Cấu hình Apache server

Giao diện giữa người quản trị và chương trình trên máy CA được thực hiện thông qua trình duyệt Netscape, do vậy sau khi cài đặt phần mềm CA để chương trình hoạt động cần thiết lập cấu hình cho chương trình CA trên Apache. Việc thiết lập cấu hình để CA sử dụng Apache được tiến hành như sau:

Trong tệp cấu hình của Apache (tệp httpd.conf trong thư mục /ect/httpd/conf) cần bổ sung trang giao diện CA trong mục "VirtualHost" như sau:

```
<VirtualHost 200.1.1.2>
    DocumentRoot "/home/myca/cgi-print/"
    ServerName printcert
    Errorlog logs/print/error_log
    CustomLog logs/print/access_log common
    ScriptAlias /cgi-bin/ "/home/myca/cgi-print/"
    <Directory "/home/myca/cgi-print">
      AllowOverride None
      Options ExecCGI
      Order allow,deny
      Allow from all
    </Directory>
</VirtualHost>

<VirtualHost 200.1.1.2>
    DocumentRoot "/home/myca/htdocs-ca/"
    ServerName rootca
    Errorlog logs/ca/error_log
    CustomLog logs/ca/access_log common
    ScriptAlias /cgi-bin/ "/home/myca/cgi-ca/"
    <Directory "/home/myca/cgi-ca">
      AllowOverride None
      Options ExecCGI
      Order allow,deny
      Allow from all
    </Directory>
</VirtualHost>
```

Trong đó trang printcert được sử dụng để in giấy chứng nhận cấp chngs chỉ số cho người sử dụng, và trang rootca là giao diện chính để người quản trị thực hiện việc phát hành hủy bỏ chứng chỉ.

- Trong tệp /etc/hots bổ sung thêm các trang trên:

  200.1.1.2 rootca printcert

- Cần tạo các thư mục: ca, print trong /etc/httpd/logs để lưu lại nhật ký, thông báo lỗi nếu chương trình xuất hiện lỗi.
- Sau khi thực hiện cấu hình xong cần khởi động lại Apache để các tham số mới được bổ sung có hiệu lực, bằng cách thực hiện lệnh sau:

  /etc/init.d/httpd restart.

Cấu hình cho MySSL và MyCA

Tất cả các tham số cấu hình cho trình MySSL, MyCA tương ứng được để trong các tệp sau /MyCA/conf/myssl.cnf và /home/htppd/cgi-ca/ca.conf. Hầu hết các tham số trong hai tệp này có thể dùng chung cho tòn hệ thống, tuy nhiên trong đó có những tham số mà đối với mỗi máy CA (cả root hoặc nonroot) cần có sự thay đổi khi chúng được thiết lập.

Khi một máy CA được thiết lập, cần có một cặp khóa được sinh theo số ID đã được hệ thống chấp nhận, khi đó số ID dưới dạng thập phân sẽ được dùng làm phần chính của tên tệp khóa cũng như tên tệp chứng chỉ của CA đó (giả sử CA được cấp ID là 01 thì khi khởi tạo cho CA đó tệp khóa sẽ là 01.key, tệp chngs chỉ là 01.crt). Khi đó trong tệp cấu hình của MySSL (myssl.cnf) và MyCA (ca.conf) cần thay đổi các tham số sau:

- Trong tệp myssl.cnf vào phần [CA-defaut] thay đổi hai thuộc tính chứng chỉ và private_key thành:

```
certificate = $dir/01.crt
private_key=$dir/private/01.key
```

- Tương tự trong tệp ca.conf cần thay đổi hai thuộc tính cacert và cakey và thuộc tính địa chỉ của máy public database server:

```
cacert "/MyCA/01.crt"
cakey "/MyCA/private/01.key"
ldapserver 200.1.1.1
```

Chương 4: QUY TRÌNH CẤP PHÁT CHỨNG CHỈ SỐ

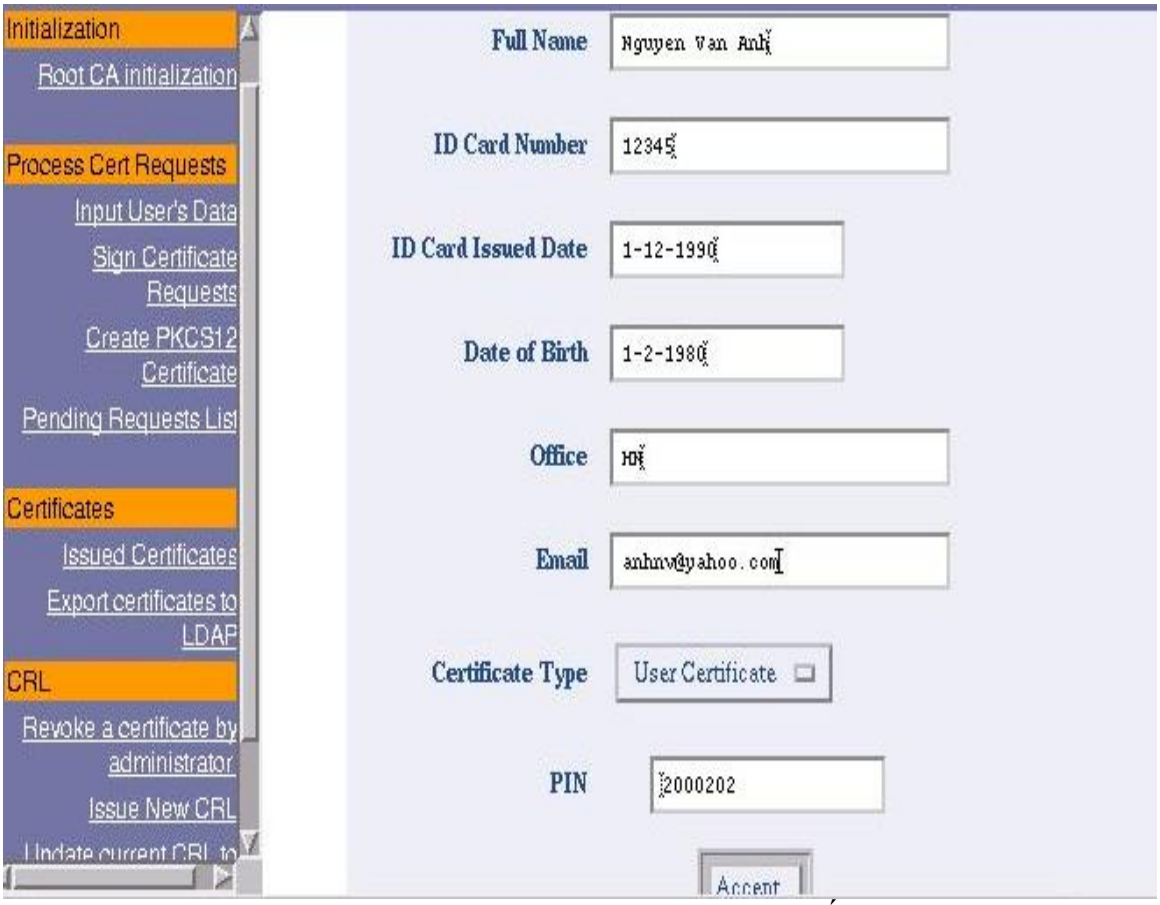
Phần mềm cấp chứng chỉ số chạy trên môi trường Linux 7.2, giao diện thực hiện các thao tác cấp chứng chỉ được thực hiện thông qua trình duyệt Web.

Hình 4.1 Giao diện phầm mềm cung cấp chứng chỉ số

Để sinh một chứng chỉ số cho một người sử dụng, chúng ta chỉ cần thực hiện ba chức năng trên giao diện chính của phần mềm, đó là: Input User's Data, Sign Certificate Requests và Generate PKCS12 Certificate. Dưới đây lần lượt là các bước thực hiện việc cấp một chứng chỉ số.

Bước 1: Nhập thông tin về người được cấp (Input User's Data)

Khi chọn chức năng này trên màn hình xuất hiện giáo diện như hình 4.2



Hình 4.2 Giao diện nhập thông tin người được cấp chứng chỉ

Người thực hiện lần lượt nhập các thông tin của người được cấp chứng chỉ vào các mục trên giao diện.

- Họ và tên (Fullname)
- Số chứng minh nhân dân (ID Card Number)
- Ngày cấp chứng minh (ID Card Issued Date)
- Ngày tháng năm sinh (Date Of Birth)
- Phòng ban (Office)
- Địa chỉ Email (Email)
- Chức năng của chứng chỉ được cấp (Certificate Type), đối với ứng dụng Mail kiểu chứng chỉ bao giờ cũng phải chọn là "User Certificate".
- Số PIN, số PIN sẽ tự động được tăng ên khi thông tin một người được chấp nhận.

Sau khi nhập đầy đủ các thông tin trên, người thực hiện chọn nút lệnh "Accept". Khi đó trên màn hình xuất hiện hộp hội thoại như hình 4.3

Hình 4.3 Giao diện cảnh báo khi nhập xong thông tin người được cấp chứng chỉ

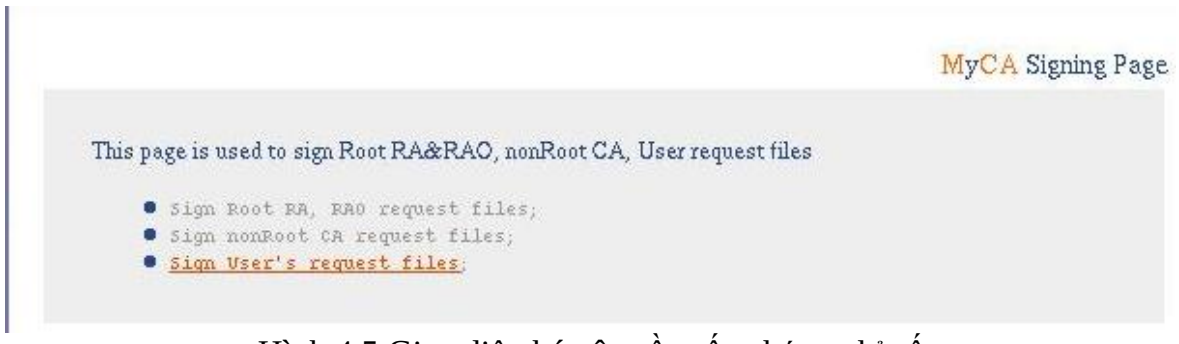
Chương trình sẽ tự động sinh yêu cầu cấp chứng chỉ số (Certificate Requests) với các thông tin trên. Quá trình sinh yêu cầu kết thúc khi trên màn hình xuất hiện thông báo như hình 4.4

Hình 4.4 Giao diện thông báo hoàn thành nhập thông tin người được cấp chứng chỉ

Sau khi thực hiện xong bước 1, trong thư mục /MyCA/user sẽ xuất hiện thêm thư mục mang tên là số ID của người sử dụng, trong đó có lưu tệp khóa bí mật và tệp yêu cầu cấp chứng chỉ của người sử dụng dưới định dạng PKCS#10.

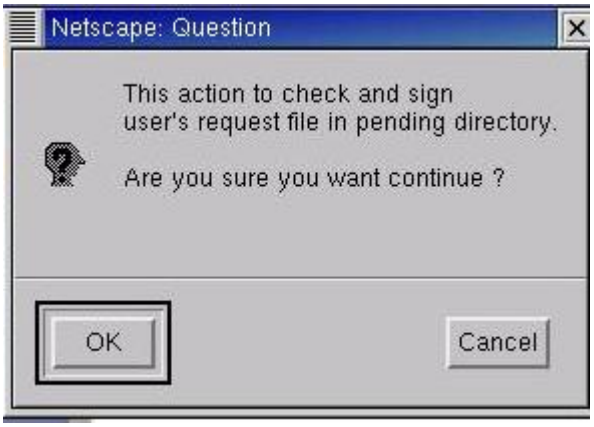
Bước 2: Ký yêu cầu cấp chứng chỉ số (Sign Certificate Requests)

Khi chọn chức năng này trên màn hình xuất hiện giao diện như hình 4.5



Hình 4.5 Giao diện ký yêu cầu cấp chứng chỉ số

Người thực hiện chọn chức năng “Sign User’s Requests Files”, khi đó trên amnf hình xuất hiện hộp hội thoại như hình 4.6

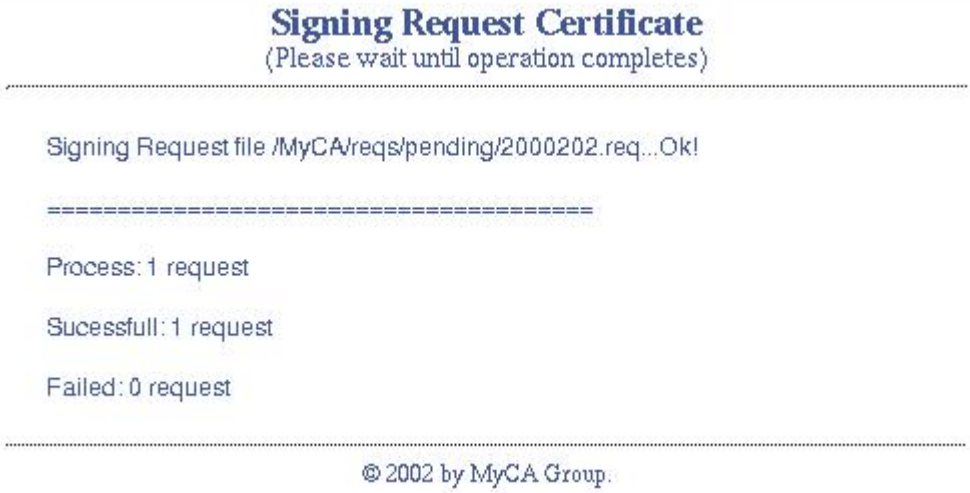


Hình 4.6 Giao diện hộp hội thoại

Người quản trị chọn “OK”, trên màn hình xuất hiện hộp hội thoại như hình 4.7

Hình 4.7 Giao diện nhập mật khẩu để giải mã khóa bí mật của CA

Người sử dụng nhập mất khẩu dùng để giải mã khóa bí mật của CA (mật khẩu này được đặt khi thực hiện thiết lập hệ thông), rồi chọn “OK”. Quá trình phát hành chứng chỉ số cho người sử dụng sẽ được thực hiện.



Hình 4.8 Giao diện thông báo cấp phát chứng chỉ thành công

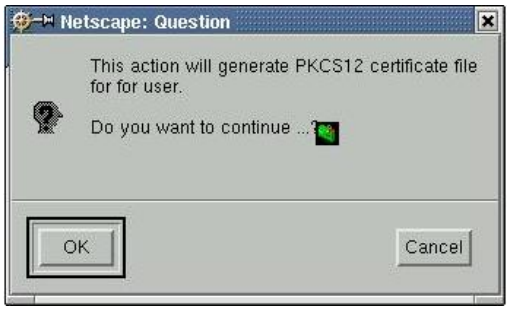
Trong ví dụ trên người được cấp chứng chỉ số có số PIN là 2000202. Quá trình phát hành chứng chỉ thành công khi có thông báo "OK!" (như ở hình 4.8), việc phát hành à không thành công nếu thay bởi thông báo "OK!" chương trình thông báo "Failed!".

Bước 3: Chuyển đổi định dạng của chứng chỉ (Generate PKCS12 Certificate)

Sau khi đã phát hành chứng chỉ số, để cài đặt được chứng chỉ cho ứng dụng Mail hoặc lưu vào thiệt bị IKEY, thì chứng chỉ số cần được chuyển đổi định dàng thành dạng PKCS12, để thực hiện sử dụng chức năng "Generate PKCS12 Certificate", khi đó trên màn hình xuất hiện giao diện như hình 4.9.

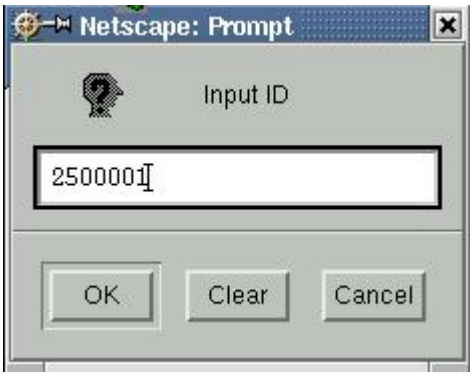
Hình 4.9 Giao diện chuyển đổi định dạng PKCS10 thành PKCS12

Người thực hiện chọn "Generate User's PKCS12 files", trên màn hình xuất hiện hộp hội thoại như hình 4.10.



Hình 4.10 Giao diện thông báo khi chuyển đổi PKCS12

Người thực hiện chọn "OK", trên màn hình xuất hiện hộp hội thoại như hình 4.11

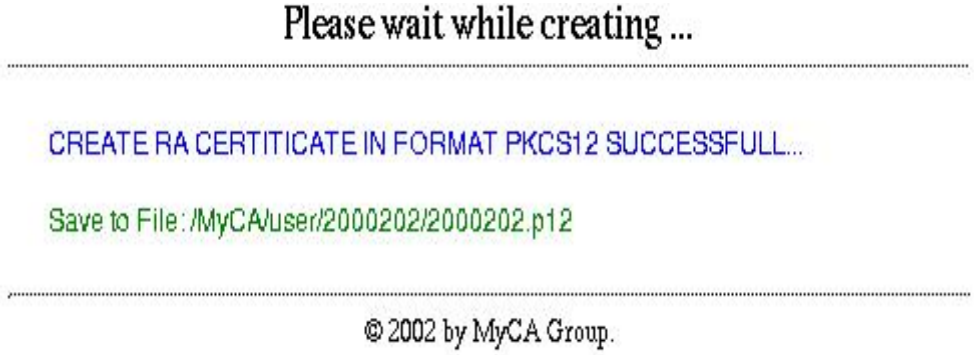


Hình 4.11 Giao diện nhập số PIN

Người thực hiện nhập số PIN của người được cấp chứng chỉ rồi chọn "OK", trên màn hình xuất hiện hộp hội thoại như hình 4.12

Hình 4.12 Giao diện nhập mật khẩu mã hóa

Người quản trị nhập mật khẩu mã hóa khóa bí mật trong tệp PKCS#12 rồi chọn "OK", quá trình chuyển đổi được thực hiện như thông báo trên màn hình

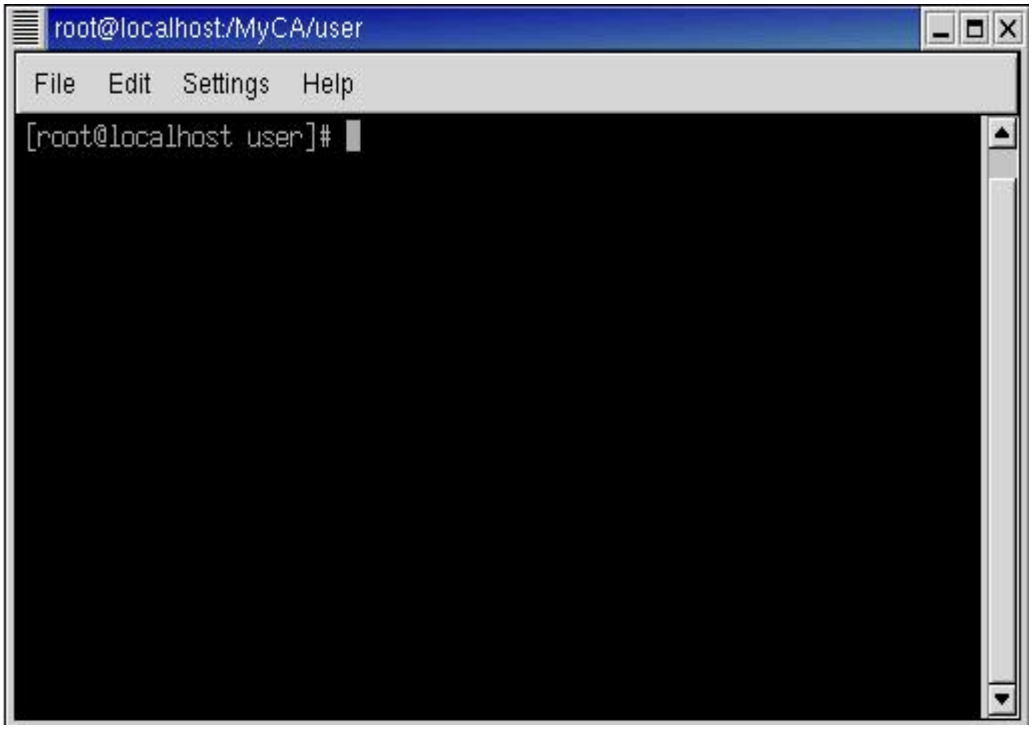


Hình 4.13 Giao diện thông báo chuyển đổi thành công

Quá trình sinh chứng chỉ kết thúc.

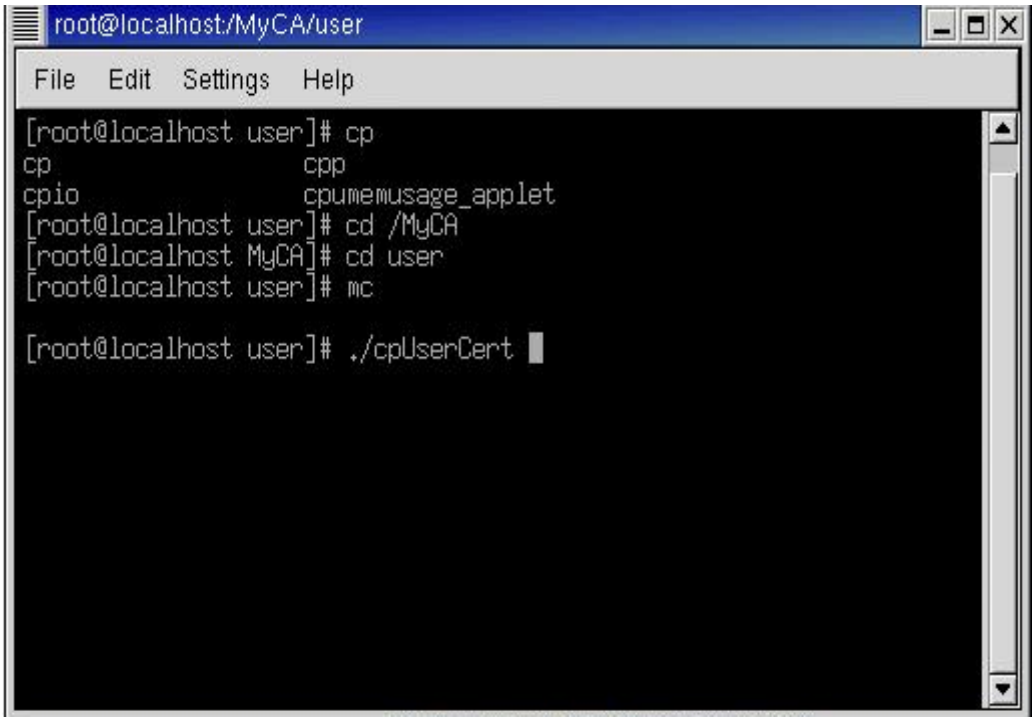
Bước 4: Cấp chứng chỉ cho người dùng

Bản chất của bước này à copy chứng chỉ vào đĩa mềm cho người sử dụng. Để thực hiện mở màn hình commandline, chuyển thư mục hiện hành thành /MyCA/user, như hình 4.14.



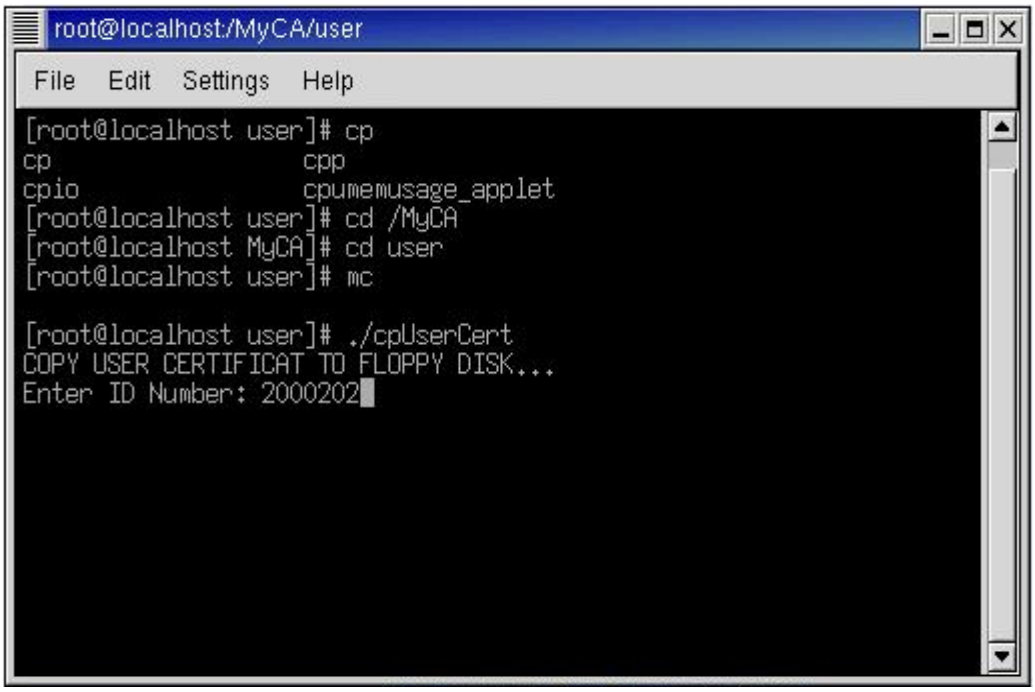
Hình 4.14 Giao diện màn hình commandline

Cho đĩa mềm vào ổ và thực hiện lệnh "./copyUserCert" như hình 4.15



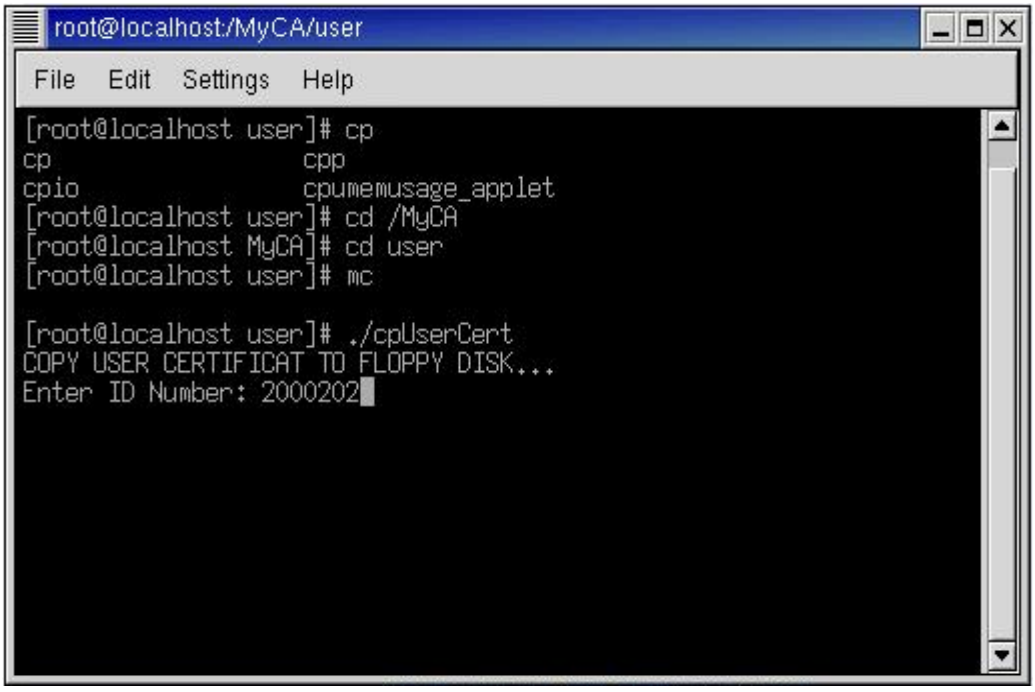
Hình 4.15 Giao diện thực thi lệnh copyUserCert

Người thực hiện nhấp số PIN của người sử dụng cần copy chứng chỉ số như hình 4.16



Hình 4.16 Giao diện nhập số PIN

Quá trình copy chứng chỉ số của người sử dụng và chứng chỉ của CA lên đĩa mềm được thực hiện và thông báo như trên hình 4.17



Hình 4.17 Giao diện thông báo hoàn thành cấp chứng chỉ

Quá trình cấp chứng chỉ số kết thúc. Người sử dụng được cấp một đĩa mềm trên đó có chứng chỉ số của họ dưới định dạng PKCS12 và chứng chỉ của CA. Người sử dụng sẽ thực hiện cài đặt các chứng chỉ này cho ứng dụng Mail.

Bước 5: Cập nhật chứng chỉ vừa phát hành lên DAP server

Để thực hiện, người quản trị chọn chức năng "Export Certificates to LDAP server", khi đó trên màn hình xuất hiện thông báo như hình 4.18

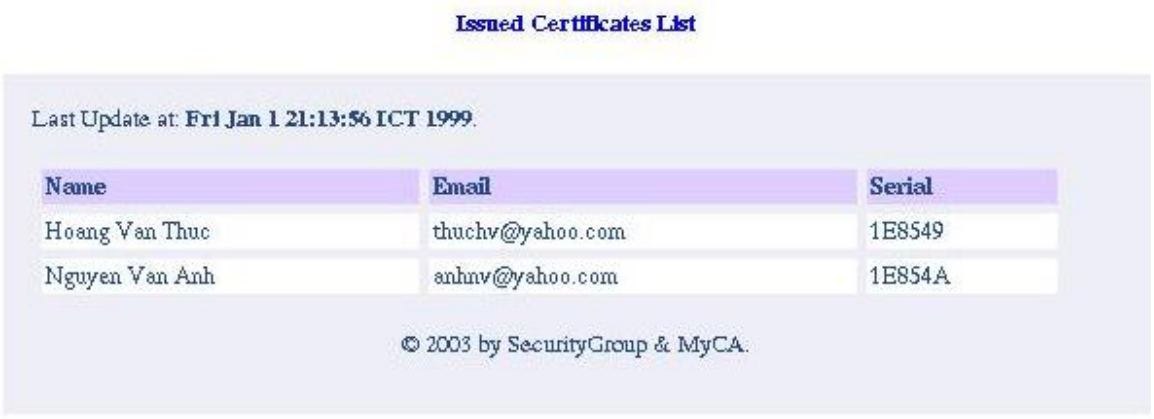
Hình 4.18 Giao diện thông báo cập nhật chứng chỉ

Ngoài các chức năng trên, trên giao diện chính còn hai chức năng nữa nhưng đây chỉ là các chức năng phụ không cần quan tâm.

Chức năng "Pending Request List": hiển thị các yêu cầu chưa được ký. Khi chọn chức năng này trên màn hình xuất hiện danh sách các request chưa được ký như hình 4.19.

Hình 4.19 Giao diện chức năng "Pending Request List"

Chức năng "Issue Certificate": hiển thị danh sách các chứng chỉ đã cấp như hình 4.20



Hình 4.20 Giao diện chức năng "Issue Certificate"

Bước 6: In nội dung chứng chỉ

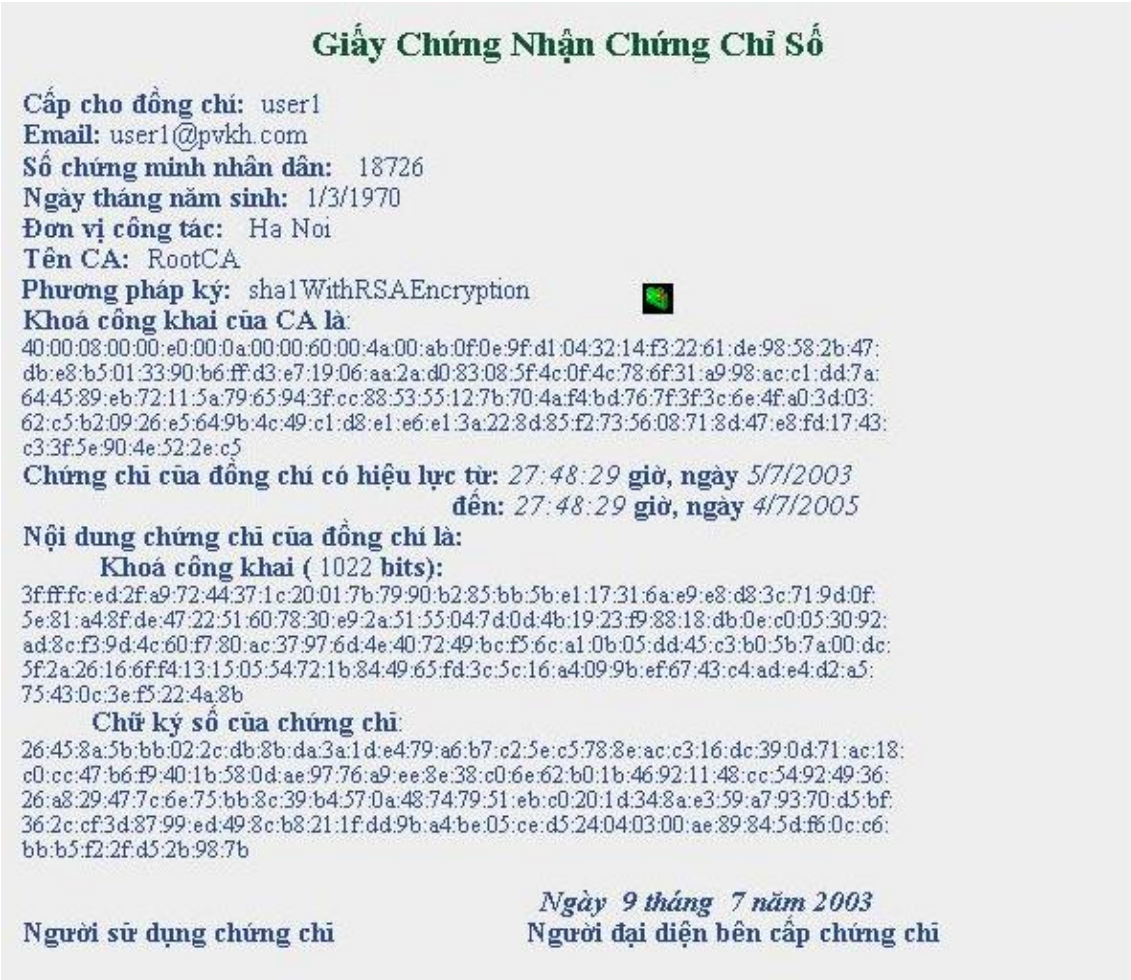
Sử dụng trang http://printcert khi đó trên màn hình Netscape xuất hiện giao diện như hình 4.21

Hình 4.21 Giao diện trang Printcert

Nhấp chuột vào "Print Certificate", trên màn hình xuất hiện form như hình 4.22

Hình 4.22 Giao diện form nhập số PIN của chứng chỉ

Người thực hiện gõ số PIN của chứng chỉ cần in, rồi nhấn "Continue...", trên màn hình xuất hiện nội dung cần in như hình 4.23



Hình 4.23 Giao diện giấy chứng nhận chứng chỉ số

Vào menu File của trình duyệt Netscape, chọn chức năng Print để in nội dung của chứng chỉ cấp cho người sử dụng.

KẾT LUẬN

Với đề tài "Tìm hiểu Hệ thống cung cấp chứng chỉ số theo mô hình sinh khóa tập trung". Em đã mang những kiến thức được học ở nhà trường đem vận dụng vào thực tế để xây dựng bài toán này. Qua đó em có điều kiện trau dồi, nâng cao kiến thức đã học. Đồ án này cũng cho em bước đấu làm quen với công tác báo mật, hiểu thêm về cách quán lý chứng chỉ số.

Với yêu cầu của bài toàn về cấn đề tìm hiểu hệ thống cung cấp chứng chỉ số theo mô hình sinh khóa tập trung thì bước đầu em đã đạt được một số kết quả sau:

- Nắm được kiến thức về mật mã hóa công khai, chữ ký số và chứng chỉ số.
- Xây dựng được chương trình tạo chữ ký số và xác thực chữ ký số.
- Tuy nhiên em vẫn chưa xây dựng được mô hình cấp phát chứng chỉ số.

Hướng phát triển tiếp theo em sẽ xây dựng mô hính cấp phát chứng chỉ số với các tính năng cung cấp, sửa đổi và xóa.

Do kiến thức còn hạn chế nên đồ án tốt nghiệp của em chắc chắn không tránh khỏi những thiếu sót. Em rất mong có được những ý kiến đánh giá, đóng góp của các thày cô và các bạn để nội dung đồ án thêm hoàn thiện.

TÀI LIỆU THAM KHẢO

Tài liệu Tiếng Việt:

[1] Hồ Văn Canh, Nguyễn Viết Thế: Nhập môn phân tích thông tin có bảo mật, NXB Hà Nội T&T, 4/2010.

[2] Trịnh Nhật Tiến. Nghiên cứu xây dựng cơ sở hạ tầng khóa công khai (PKI) đảm bảo an toàn truyền tin trên mạng máy tính (Đề tài cấp Thành phố đã được nghiệm thu - 2005).

[3] Phan Đình Diệu: An toàn thông tin và mật mã, NXB Trường Đại học Quốc gia Hà Nội, 2002.

[4] Nguyễn Thế Dân: Một số vấn đề triển khai chứng chỉ điện tử tại Việt Nam.

[5] Nguyễn Nam Hải, Đào Thị Hồng Vân: Chứng thực trong thương mại điện tử, NXB KHKT năm 2004.

Tài liệu Tiếng Anh:

[1] Andrew Nash, William Duane, Celia Joseph, Derek Brink (2001): Public Key Infrastructure and Its Application.

[2] Implementing and Managing E-Securing, NXB McGraw-Hill.