Applied Cryptography - Lab Documentation

Chapter 14: Key Management and Distribution

Thai Tan Tran - Chi Thong Thach - Huu Thuan Nguyen

2024-20-03

Contents

1 Annonation

Name	Notation	Description
Master Key	K	Key used for a long-term communication
Session Key	K_s	Key used for a single communication session
Key Distribution Center	KDC	A trusted entity that distributes keys
Identity	ID	A unique identifier for a user or entity
Nonce	N	A number used once to prevent replay attack
Control Vector	CV	A vector used to control key usage

2 Symmetric Key

Mã hoá đối xứng là một phương pháp mã hoá thông tin sử dụng cùng một key để mã hoá và giải mã. Nó hiệu quả hơn so với mã hoá không đối xứng đối với các dữ liệu lớn.

Việc phân phối key cho hai bên Alice và Bob có thể được thực hiện theo 2 cách sau:

1. Alice chọn một key và trực tiếp giao cho Bob.
2. Một bên thứ ba (gọi là KDC) chọn và phân phối key cho cả Alice và Bob.

Tuy nhiên cả 2 cách trên đều không phù hợp vì việc phân phối key thủ công và trực tiếp như vậy là không hiệu quả và không an toàn.

Chúng ta sẽ giải quyết vấn đề này bằng cách sử dụng end-to-end encryption như 2 phiên bản cải tiến sau:

1. Nếu Alice và Bob gần đây dùng một key, một bên có thể truyền key mới cho bên kia bằng cách mã hóa nó bằng key cũ.
2. Nếu mỗi Alice và Bob đều có một encrypted connection với KDC, KDC có thể chuyển giao key đã được masked bởi một lớp encryption cho Alice và Bob.

Điều này là an toàn hơn, tuy nhiên khi xét đến một hệ thống như vậy, mỗi thiết bị đều cần một lượng key cung cấp động. Đây là một vấn đề khó trong hệ thống có phạm vi lớn.

Cụ thể, một người khi muốn giao tiếp với n người khác nhau sẽ cần n key để mã hoá thông tin đối với mỗi người nhận. Do đó, trong trường hợp n người cần giao tiếp với n người khác sẽ cần đến $n*(n-1)$ key.

Do đó, chúng ta cần một cơ chế hiệu quả cho bên thứ ba là KDC để đảm bảo việc phân phối key là an toàn, hiệu quả và dễ dàng.

2.1 Symmetric Key Distribution using Symmetric Encryption

2.1.1 Solution

Một trong những giải pháp cho KDC là dựa trên việc sử dụng một hệ thống phân cấp cho các key (gọi là *hierarchical key control*). Ở mô hình đơn giản nhất sẽ có 2 cấp độ key được sử dụng, đó là *session key* và *master key*.

Trong giải pháp này, việc giao tiếp giữa các bên được mã hoá bởi một key tạm thời, gọi là *session key*. *Session key* thường được sử dụng cho mỗi lần kết nối và được loại bỏ khi không cần thiết.

Mỗi *session key* sẽ được lấy từ KDC qua cùng một cơ sở mạng. Do đó, *session key* được truyền đi trong dạng mã hóa, lúc này, chúng ta sẽ cần đến *master key* để giải mã.

Master key là key mà mỗi user chia sẻ với KDC trước khi bắt đầu giao tiếp để đảm nhiệm việc mã hoá và giải mã sau này. Do đó, chỉ có duy nhất một *master key* được chia sẻ với KDC. Dẫn đến quy mô sẽ được giảm đáng kể, nếu có n user thì chỉ cần $\frac{n(n-1)}{2}$ *session key* vào bất kì thời điểm nào theo lý thuyết, nhưng chỉ cần n *master key*.

2.1.2 Scenario

Giả sử rằng Alice muốn kết nối với Bob và yêu cầu một *session key* để bảo vệ dữ liệu khi truyền. Alice có *master key* K_a chỉ có KDC biết. Bob tương tự với *master key* K_b.

1. Alice gửi yêu cầu đến KDC để nhận *session key* bảo vệ kết nối với Bob.
2. KDC phản hồi Alice và Bob bằng một ciphertext. Nó bao gồm *session key* K_s được mã hoá bởi K_a cho Alice và K_b cho Bob.

Tất nhiên, trong quá trình yêu cầu và phản hồi, sẽ cần có ID_a và ID_b để xác định danh tính của Alice và Bob.

Lúc này, một *session key* đã được phân phối bảo mật cho Alice và Bob và họ có thể bắt đầu trao đổi thông tin một cách an toàn. Tuy nhiên, có thể xem xét 2 bước bổ sung sau:

1. Bob sử dụng *session key* K_s để mã hóa và gửi một nonce N đến Alice.
2. Alice sử dụng *session key* K_s để phản hồi Bob với $f(N)$, nơi f là một hàm thực hiện biến đổi trên N (ví dụ cộng thêm 1).

Các bước này đảm bảo với Bob rằng message ban đầu mà nó nhận được (bước 1) không phải là message được gửi lại. Do đó, trong giải pháp này, sẽ có một yếu tố bổ sung gọi là nonce N giúp ngăn chặn cuộc tấn công phát lại (replay attack).

2.1.3 Improvement

Không cần thiết phải giới hạn chức năng phân phối key vào một KDC duy nhất, đặc biệt đối với mạng lớn. Thay vào đó, có thể thiết lập một hệ thống phân cấp các KDC.

Hệ thống phân cấp có thể mở rộng lên nhiều tầng tùy thuộc vào quy mô người dùng và phạm vi địa lý của mạng. Hệ thống phân cấp giảm thiểu công sức phân phối key và giới hạn rủi ro khi có lỗi hoặc bị tấn công, vì key chủ chỉ được chia sẻ trong phạm vi local KDC.

2.1.4 Problem

Thay đổi *session key* thường xuyên làm tăng an ninh vì kẻ tấn công có ít ciphertext để phân tích, nhưng việc này gây áp lực lên hiệu suất của KDC. Do đó chúng ta cần phải cân nhắc giữa an ninh và hiệu suất để quyết định thời gian tồn tại của một *session key*.

Trong *connection-oriented protocol*, việc sử dụng cùng một *session key* suốt thời gian kết nối là lựa chọn hiển nhiên, nhưng với các kết nối dài hạn, cần phải thay đổi key định kỳ.

Đối với *connectionaless protocol*, không có sự khởi tạo hay kết thúc rõ ràng cho mỗi kết nối, nên việc thay đổi key không rõ ràng.

Lối tiếp cận an toàn nhất là sử dụng key mới cho mỗi lần giao tiếp, nhưng điều này lại mâu thuẫn với lợi ích của *connectionaless protocol* là tối thiểu hóa độ trễ và tải hệ thống. Do đó, chiến lược tốt hơn là sử dụng một *session key* cho một khoảng thời gian cố định hoặc cho một số lượng kết nối cố định.

2.1.5 A Transparent Key Control Scheme

Phương pháp được mô tả cho việc thiết lập kết nối và mã hóa đầu cuối thông qua một giao thức hướng kết nối như TCP bao gồm các bước sau:

1. Máy chủ gửi yêu cầu kết nối tới máy chủ khác.
2. Session security module (SSM) lưu yêu cầu và xin phép KDC để thiết lập kết nối.
3. KDC, sau khi xác minh và chấp thuận yêu cầu kết nối, tạo ra session key và gửi nó đến hai SSM phù hợp sử dụng một khóa vĩnh viễn duy nhất cho mỗi SSM.
4. SSM yêu cầu phát hành connection request package, thiết lập kết nối giữa hai hệ thống và tất cả dữ liệu trao đổi sau đó đều được mã hóa bởi SSM sử dụng khóa phiên.

Hệ thống phân phối khóa tự động này cung cấp tính linh hoạt và đặc tính động cần thiết để cho phép nhiều người dùng thiết bị đầu cuối truy cập nhiều máy chủ và cho phép các máy chủ này trao đổi dữ liệu với nhau. Điều này đảm bảo rằng việc mã hóa đầu cuối diễn ra mà người dùng cuối không nhận thức được, đảm bảo tính minh bạch và an toàn trong quá trình giao tiếp.

2.1.6 Decentralized Key Control

Trong quản lý khóa phi tập trung, mỗi node mạng cần có khả năng giao tiếp an toàn với mọi node khác và có thể cần đến $\frac{n-1}{2}$ master key cho n node mạng. Quá trình thiết lập session key diễn ra như sau:

1. Nút A yêu cầu session key từ nút B, kèm theo một nonce N_1.
2. Nút B phản hồi với thông điệp được mã hóa bằng khóa chủ chia sẻ, bao gồm session key mà B chọn, ID của B, hàm f của N_1 và nonce mới N_2.
3. A sử dụng session key mới để trả lời B với $f(N_2)$.

Mỗi node chỉ cần duy trì tối đa $n-1$ master key và có thể tạo ra nhiều session key khi cần. Do thông điệp được truyền đi sử dụng master key ngắn, phân tích mật mã trở nên khó khăn. Session key được sử dụng trong thời gian hạn chế để bảo vệ chúng.

2.1.7 Controlling Key Usage

Hệ thống phân cấp khóa và kỹ thuật phân phối khóa tự động làm giảm số lượng khóa cần quản lý thủ công, làm tăng hiệu quả quản lý khóa. Hệ thống còn hỗ trợ kiểm soát việc sử dụng khóa thông qua quy tắc và hạn chế, nâng cao an ninh và giảm rủi ro. Chẳng hạn, ngoài việc phân biệt master key từ session key, chúng ta có thể muốn xác định các loại session key khác nhau dựa trên cơ sở sử dụng, như:

- Data-encrypting key, dùng cho việc giao tiếp chung qua mạng
- PIN-encrypting key, dùng cho các số nhận dạng cá nhân (PINs) được sử dụng trong chuyển khoản điện tử và ứng dụng thanh toán tại điểm bán hàng
- File-encrypting key, dùng để mã hóa các tệp lưu trữ ở những nơi có thể truy cập công cộng

Để ngăn chặn rủi ro từ việc sử dụng không chính xác master key như một session key, cần phải thiết lập các kiểm soát trong hệ thống. Cách tiếp cận này bao gồm việc gắn một nhãn điện tử cho mỗi khóa. Trong kỹ thuật được đề xuất cho hệ thống DES, 8 bit không sử dụng của mỗi khóa 64-bit sẽ được dùng để tạo nhãn này, với một bit xác định loại khóa (master hay session), một bit chỉ ra khả năng mã hóa, một bit cho khả năng giải mã, và các bit còn lại được giữ để mở rộng chức năng trong tương lai. Điều này giúp kiểm soát cách thức sử dụng các khóa một cách chặt chẽ hơn.

Khi nhãn được nhúng vào trong khóa, nó được mã hóa cùng với khóa khi khóa đó được phân phối, từ đó cung cấp một tầng bảo vệ. Tuy nhiên, cách tiếp cận này có hai nhược điểm chính:

1. Độ dài của nhãn giới hạn chỉ ở 8 bits, làm giảm tính linh hoạt và chức năng của nó.
2. Bởi vì nhãn không được truyền đi dưới dạng bản rõ (tức là không mã hóa), nó chỉ có thể sử dụng tại điểm giải mã, hạn chế cách thức kiểm soát việc sử dụng khóa.

Để cải thiện điều này, một cách tiếp cận linh hoạt hơn được mô tả, được gọi là "control vector". Trong cách tiếp cận này, mỗi session key được liên kết với một control vector gồm nhiều trường, mỗi trường xác định các sử dụng và hạn chế cho session key đó. Độ dài của control vector có thể thay đổi, cho phép nhiều thông tin điều khiển hơn được gắn với khóa và do đó tăng cường khả năng kiểm soát việc sử dụng khóa.

Các bước thực hiện:

1. Control vector trước tiên được chuyển qua một hàm băm để tạo ra một giá trị có độ dài bằng với độ dài của khóa mã hóa. Hàm băm biến đổi các giá trị từ một phạm vi lớn sang một phạm vi nhỏ hơn với sự phân bố đều.
2. Giá trị băm sau đó được XOR với master key để tạo ra một đầu ra, và đầu ra này được sử dụng làm đầu vào khóa để mã hóa session key. Cụ thể, công thức được đưa ra như sau:
   - Giá trị băm $H = h(CV)$
   - Đầu vào khóa $K_m \oplus H$
   - Văn bản mã $E([K_m \oplus H], K_s)$ Trong đó, K_m là master key và K_s là session key. Session key có thể được phục hồi dưới dạng bản rõ bằng phép toán ngược $D([K_m \oplus H], E([K_m \oplus H], K_s))$
3. Khi một session key được giao cho người dùng từ KDC, nó đi kèm với control vector dưới dạng plaintext. Session key chỉ có thể được phục hồi bởi người dùng có cả master key mà họ chia sẻ với KDC và control vector. Do đó, liên kết giữa session key và control vector của nó được bảo toàn.

Sử dụng control vector mang lại hai lợi ích so với việc sử dụng một nhãn 8-bit:

1. Không có hạn chế về độ dài của control vector, điều này cho phép áp đặt các điều khiển phức tạp một cách tùy ý lên việc sử dụng khóa.
2. Control vector có sẵn dưới dạng plaintext ở mọi giai đoạn của quá trình vận hành. Do đó, việc kiểm soát sử dụng khóa có thể được thực hiện ở nhiều địa điểm.

Những ưu điểm này giúp tăng cường khả năng bảo mật và quản lý linh hoạt cách thức sử dụng các khóa trong hệ thống mật mã.

2.2 Symmetric Key Distribution using Asymmetric Encryption

Vì sự không hiệu quả của các public-key cryptosystems, chúng gần như không bao giờ được sử dụng để mã hóa trực tiếp các khối dữ liệu có kích thước lớn, nhưng chỉ giới hạn ở các khối nhỏ tương đối. Một trong những ứng dụng quan trọng nhất của public-key cryptosystems là để mã hóa các secret key để phân phối. Simple Secret Key Distribution

Một sơ đồ đơn giản được được đề xuất bởi Merkle và một kịch bản tấn công man-in-the-middle (MITM) có thể xảy ra trong giao thức này. A tạo một public/private key và gửi một thông điệp tới B bao gồm khóa công khai và ID của A, . B tạo một secret key và truyền nó tới A, được mã hóa bằng khóa công khai của A. A giải mã bằng cách dùng khóa riêng tư của mình để giải mã thông điệp đã mã hóa và phục hồi khóa bí mật. Chỉ có A mới có thể giải mã thông điệp, do đó chỉ có A và B biết khóa bí mật . A loại bỏ cặp khóa công khai và riêng tư của mình, và B cũng loại bỏ khóa công khai của A. Sau đó, A và B có thể giao tiếp an toàn bằng cách sử dụng mã hóa thông thường vóie session key . Khi giao dịch kết thúc, cả hai đều loại bỏ khóa . Mặc dù đơn giản, nhưng giao thức này khá hấp dẫn vì không có khóa nào tồn tại trước khi giao tiếp bắt đầu và cũng không có khóa nào còn lại sau khi giao tiếp kết thúc. Nhờ vậy, rủi ro bị tiết lộ khóa là rất nhỏ và giao tiếp đảm bảo an toàn trước nguy cơ nghe lén.

Mô tả một tấn công man-in-the-middle như sau: A tạo một public/private key và gửi một thông điệp cho B bao gồm khóa công khai và ID của A, . D (kẻ tấn công) chặn thông điệp này, tạo ra public/private key của mình và gửi tới B. B tạo một a secret key và gửi nó đã mã hóa với khóa công

khai mà B tin là của A . D chặn thông điệp này và biết được bằng cách giải mã thông điệp với khóa riêng của mình . D sau đó gửi tới A. Kết quả là cả A và B đều biết , và không hề hay biết rằng cũng đã bị tiết lộ cho D. A và B giờ đây có thể trao đổi thông điệp sử dụng , và D không còn can thiệp vào kênh truyền thông nữa nhưng chỉ đơn giản là nghe lén. Biết , D có thể giải mã tất cả các thông điệp, và cả A và B đều không ý thức được vấn đề. Do đó, giao thức đơn giản này chỉ hữu ích trong một môi trường nơi mối đe dọa duy nhất là việc nghe lén. Secret Key Distribution with Confidentiality and Authentication

Hình 14.9, dựa trên cách tiếp cận được đề xuất trong [NEED78], cung cấp khả năng bảo vệ chống lại cả các cuộc tấn công chủ động và thụ động. Chúng ta bắt đầu tại thời điểm giả định rằng A và B đã trao đổi khóa công khai theo một trong các sơ đồ được mô tả sau trong chương này. Sau đó xảy ra các bước sau: A sử dụng khóa công khai của B để mã hóa một thông điệp gửi đến B chứa ID của A và một nonce , được sử dụng để xác định duy nhất giao dịch này. B gửi phản hồi một thông điệp được mã hóa với khóa công khai của A và chứa nonce của A cũng như một nonce mới được B tạo ra . Chỉ có B mới có thể giải mã thông điệp được mã hóa (1), sự hiện diện của trong thông điệp (2) đảm bảo cho A rằng người trả lời chính xác là B. A gửi trả , được mã hóa bằng khóa công khai của B, để chắc chắn rằng người gửi thông điệp là A. A chọn một khóa bí mật và gửi cho B. Việc mã hóa thông điệp này với khóa công khai của B đảm bảo rằng chỉ B mới có thể đọc nó; việc mã hóa với khóa riêng tư của A đảm bảo rằng chỉ A mới có thể là người đã gửi nó. (Chữ ký số) B giải mã bằng cách sử dụng để phục hồi secret key. Kết quả là giao thức này đảm bảo cả tính bảo mật và xác thực trong quá trình trao đổi khóa bí mật. A Hybrid Scheme Một cách khác để sử dụng mã hóa khóa công khai nhằm phân phối các khóa bí mật là thông qua một cách tiếp cận lai được sử dụng trên các máy chủ lớn của IBM [LE93]. Cách tiếp cận này giữ lại việc sử dụng một trung tâm phân phối khóa (KDC) chia sẻ một khóa chủ bí mật với mỗi người dùng và phân phối các khóa phiên bí mật được mã hóa với khóa chủ. Một cách tiếp cận khóa công khai được sử dụng để phân phối các khóa chủ. Lý do sau đây được đưa ra cho việc sử dụng cách tiếp cận ba tầng này: Hiệu Suất: Có nhiều ứng dụng, đặc biệt là ứng dụng giao dịch, nơi mà các khóa phiên thay đổi thường xuyên. Việc phân phối các khóa phiên bằng mã hóa khóa công khai có thể làm giảm hiệu suất hệ thống tổng thể do tải tính toán cao của việc mã hóa và giải mã bằng mã hóa khóa công khai. Với một cấu trúc phân cấp ba tầng, mã hóa khóa công khai chỉ được sử dụng thỉnh thoảng để cập nhật khóa chủ giữa người dùng và KDC. Tương thích Ngược: Cách tiếp cận lai dễ dàng được áp dụng lên trên một hệ thống KDC hiện có với ít xáo trộn hoặc thay đổi phần mềm. Thêm vào đó, việc áp dụng một lớp khóa công khai cung cấp một phương tiện an toàn, hiệu quả để phân phối các khóa chủ. Đây là một lợi thế trong một cấu hình nơi mà một KDC duy nhất phục vụ một tập hợp người dùng rộng lớn.

3 Distribution of Public Keys

Public Announcement of Public Keys Về mặt hình thức, ý nghĩa của public-key encryption: public key là public. Do đó, nếu có một thuật toán khóa công khai nào đó được chấp nhận rộng rãi, như RSA, bất kỳ người tham gia nào cũng có thể gửi khóa công khai của họ đến toàn thể cộng đồng (hình 14.10).

Mặc dù cách tiếp cận này rất tiện lợi, nó có một nhược điểm lớn. Bất kỳ ai cũng có thể làm giả thông báo công khai đó. Ví dụ, một người dùng nào đó có thể giả mạo là người dùng A và gửi một khóa công khai đến người tham gia khác hoặc broadcast một khóa công khai như thế. Cho đến khi người dùng A phát hiện ra sự giả mạo và thông báo cho các bên tham gia khác, kẻ giả mạo có thể đọc tất cả các thông điệp được mã hóa dành cho A và sử dụng các khóa giả mạo để xác thực. Publicly Available Directory

Một mức độ bảo mật cao hơn có thể được đạt được bằng cách duy trì một publicly available dynamic directory của các public key. Việc bảo trì và phân phối public directory phải là trách nhiệm của một thực thể hoặc tổ chức đáng tin cậy nào đó (Hình 14.11). Một kế hoạch như vậy sẽ bao gồm các yếu tố sau: Cơ quan chủ quản duy trì một thư mục với mục nhập {name, public key} cho mỗi người tham

gia. Mỗi người tham gia đăng ký một public key với cơ quan chủ quản thư mục. Việc đăng ký này phải được thực hiện trực tiếp hoặc thông qua một hình thức xác thực an toàn nào đó. Một người tham gia có thể thay thế khóa hiện tại bằng một khóa mới bất cứ lúc nào, dù là vì mong muốn thay thế một public key đã được sử dụng cho một lượng lớn dữ liệu, hoặc vì private key tương ứng đã bị xâm phạm theo cách nào đó. Người tham gia cũng có thể truy cập thư mục một cách điện tử. Với mục đích này, việc giao tiếp được bảo mật, xác thực từ cơ quan chủ quản đến người tham gia là bắt buộc. Rõ ràng, kế hoạch này an toàn hơn nhiều so với việc công bố công khai cá nhân nhưng vẫn còn những điểm yếu. Nếu một kẻ tấn công thành công trong việc lấy được hoặc tính toán khóa riêng tư của cơ quan chủ quản thư mục, kẻ tấn công đó có thể phân phối các khóa công cộng giả mạo và sau đó giả mạo bất kỳ người tham gia nào và nghe lén thông điệp được gửi đến bất kỳ người tham gia nào. Một cách khác để đạt được mục tiêu tương tự đối với kẻ tấn công là can thiệp vào các bản ghi được giữ bởi cơ quan chủ quản. Public-Key Authority

Bảo mật mạnh mẽ hơn cho public-key distribution có thể đạt được bằng cách kiểm soát chặt chẽ hơn việc public key từ thư mục. Một kịch bản điển hình được mô tả trong Hình 14.12, dựa trên ý tưởng từ [POPE97]. Kịch bản giả định rằng một central authority duy trì một thư mục động của các public key và chỉ cơ quan đó biết khóa riêng tư tương ứng. Các bước sau đây (Hình 14.12) xảy ra: A gửi một thông điệp được timestamp đến cơ quan ủy quyền khóa công cộng (public-key authority) chứa yêu cầu cho public key của B. Cơ quan ủy quyền respond với một thông điệp được mã hóa bằng khóa riêng của cơ quan, . Do đó, A có thể giải mã thông điệp sử dụng public key của cơ quan ủy quyền. Như vậy, A có thể chắc chắn rằng thông điệp bắt nguồn từ cơ quan ủy quyền. Thông điệp bao gồm những nội dung sau: Public key của B, , mà A có thể sử dụng để mã hóa thông điệp gửi đến B. Yêu cầu ban đầu để A có thể kết nối response này với response trước đó và xác minh rằng response ban đầu không bị thay đổi trước khi cơ quan nhận được. Timestamp ban đầu cho phép A xác định rằng đây không phải là thông điệp cũ từ cơ quan ủy quyền chứa khóa khác ngoài khóa công cộng hiện tại của B. A lưu trữ public key của B và cũng sử dụng nó để mã hóa một thông điệp gửi đến B chứa định danh của A và một nonce , được sử dụng để xác định giao dịch này một cách độc nhất. 4, 5. B thu hồi public key của A từ cơ quan ủy quyền theo cùng một cách mà A thu hồi public key của B. Tại thời điểm này, public key đã được gửi an toàn đến A và B, và họ có thể bắt đầu trao đổi thêm hai bước nữa như sau. B gửi một thông điệp đến A được mã hóa bằng và chứa nonce của A cũng như một nonce mới được B tạo ra . Chỉ có B mới có thể giải mã thông điệp này (3), sự có mặt của trong thông điệp (6) đảm bảo cho A rằng đối tác giao tiếp là B. A trả lại , được mã hóa bằng public key của B, để xác nhận đối tác giao tiếp của mình là B. Như vậy, tổng cộng có bảy thông điệp được yêu cầu. Tuy nhiên, năm thông điệp ban đầu có thể chỉ cần sử dụng ít hơn vì cả A và B có thể lưu public key của đối phương để sử dụng sau này - caching. Định kỳ, một người dùng nên yêu cầu public key mới của các bên tham gia khác để đảm bảo an toàn, mặc dù các bản ghi đã lưu trước đó cũng nên được xem xét kỹ lưỡng. Một cách khác để đạt được cùng một mục tiêu là can thiệp vào các bản ghi được giữ bởi cơ quan ủy quyền. Public-Key Certificates Kịch bản của Hình 14.12 rất tốt, nhưng nó cũng có một số nhược điểm. Cơ quan khóa công cộng có thể tạo thành một điểm nghẽn trong hệ thống, bởi người dùng phải yêu cầu cơ quan ủy quyền cấp một public key cho mọi người dùng khác mà họ muốn liên lạc. Như đã đề cập phần trước, thư mục {name, public key} được duy trì bởi cơ quan ủy quyền có thể bị can thiệp. Một phương pháp thay thế, lần đầu tiên được đề xuất bởi Kohnfelder [KUHN78], là sử dụng chứng chỉ (certificates) mà các bên tham gia có thể trao đổi khóa mà không cần liên lạc với cơ quan khóa công cộng, một cách đáng tin cậy như thể họ được cấp trực tiếp từ một cơ quan khóa công cộng. Về cơ bản, một chứng chỉ bao gồm một public key, một định danh của người sở hữu khóa, và toàn bộ khối được ký bởi một bên thứ ba đáng tin cậy. Bên thứ ba này thường là một cơ quan chứng chỉ được cộng đồng người dùng tin cậy, như một cơ quan chính phủ hoặc một tổ chức tài chính. Người dùng có thể trình bày public key của mình cho cơ quan ủy quyền một cách an toàn và nhận một chứng chỉ. Người dùng sau đó có thể công bố chứng chỉ. Bất kỳ ai cần public key của người dùng này có thể lấy chứng chỉ và xác minh tính xác thực của nó thông qua chữ ký đáng tin cậy đi kèm. Một người tham gia cũng có thể truyền đạt thông tin khóa của mình cho người khác bằng cách truyền chứng chỉ của họ. Người tham gia khác có thể xác minh chứng chỉ được tạo bởi cơ quan ủy

quyền. Xem xét quy trình sau: Bất kỳ người tham gia nào cũng có thể đọc một chứng chỉ để xác định tên và public key của chủ sở hữu chứng chỉ. Bất kỳ người tham gia nào cũng có thể xác minh chứng chỉ xuất phát từ cơ quan chứng chỉ và không phải là hàng giả. Chỉ có cơ quan chứng chỉ mới có thể tạo và cập nhật chứng chỉ. Denning [DENN83] thêm vào một yêu cầu bổ sung: Bất kỳ người tham gia nào cũng có thể xác minh thời gian hiệu lực của chứng chỉ.

Một certificate scheme được minh họa trong Hình 14.13. Mỗi người tham gia yêu cầu cho cơ quan chứng chỉ, cung cấp một public key và yêu cầu một chứng chỉ. Ứng dụng phải được thực hiện trực tiếp hoặc thông qua một số hình thức giao tiếp xác thực an toàn. Đối với người tham gia A, cơ quan chứng chỉ cung cấp một chứng chỉ dưới dạng sau: trong đó là khóa riêng được cơ quan sử dụng và T là timestapm. A có thể sau đó chuyển chứng chỉ này cho bất kỳ người tham gia nào khác, người đọc và xác minh chứng chỉ như sau:
Người nhận sử dụng public key của cơ quan, , để giải mã chứng chỉ. Bởi vì chứng chỉ chỉ có thể đọc được bằng public key của cơ quan, điều này xác minh rằng chứng chỉ đến từ cơ quan chứng chỉ. Các yếu tố và cung cấp cho người nhận tên và public key của chủ sở hữu chứng chỉ. Timestamp T xác nhận tính hiện hành của chứng chỉ. Timestamp đối phó với kịch bản sau đây: Khóa riêng của A bị một kẻ địch biết được. A tạo một cặp khóa mới/riêng và áp dụng cho chứng chỉ mới từ cơ quan chứng chỉ. Trong khi đó, kẻ tấn công sử dụng lại chứng chỉ cũ để gửi đến B. Nếu B sau đó mã hóa thông điệp sử dụng khóa công cộng cũ bị xâm phạm, kẻ tấn công có thể đọc những thông điệp đó. Trong bối cảnh này, việc một khóa riêng bị lộ tương tự như việc mất một thẻ tín dụng. Chủ sở hữu hủy số thẻ tín dụng nhưng vẫn có nguy cơ cho đến khi tất cả các bên giao tiếp biết rằng thẻ tín dụng cũ không còn giá trị. Do đó, dấu thời gian có tác dụng giống như một ngày hết hạn. Nếu một chứng chỉ đủ cũ, nó được coi là hết hạn. Một lược đồ đã trở nên được chấp nhận rộng rãi cho việc định dạng chứng chỉ khóa công cộng: tiêu chuẩn X.509. Chứng chỉ X.509 được sử dụng trong hầu hết các ứng dụng bảo mật mạng, bao gồm bảo mật IP, bảo mật lớp vận chuyển (TLS), và S/MIME.

4 X.509