Mật mã và An ninh mạng

Chương 5:
Quản lý khóa và xác thực người dùng

Chương 5: Quản lý khóa và xác thực người dùng

5.1 Quản lý và phân phối khóa

5.2 Xác thực người dùng

5.1 Quản lý và phân phối khóa

1. Phân phối khóa đối xứng dùng mật mã đối xứng

- Với mật mã đối xứng, 2 bên phải trao đổi cùng 1 khóa như nhau và phải bảo vệ khóa này từ những người khác.
- Cần phải thay đổi khóa thường xuyên để hạn chế lượng dữ liệu bị tổn thương nếu như kẻ tấn công tìm hiểu khóa.
- Vì vậy cần có kỹ thuật phân phối khóa để phát đi khóa cho 2 bên trao đổi dữ liệu và không cho những người khác biết khóa.

1. Phân phối khóa đối xứng dùng mật mã đối xứng

- Trung tâm phân phối khóa dựa trên việc phân cấp khóa.
- Truyền tin giữa 2 đầu cuối được mã hóa sử dụng khóa tạm thời gọi là khóa phiên (session key). Khóa phiên được dùng trong khoảng thời gian kết nối logic và sau đó phá hủy luôn.
- Khóa phiên được truyền đi dưới dạng mã hóa sử dụng khóa chủ (master key). Khóa chủ được chia sẻ bởi trung tâm phân phối khóa và một đầu cuối. Khóa chủ ít được dùng và tồn tại lâu.

1. Phân phối khóa đối xứng dùng mật mã đối xứng

a) Kịch bản

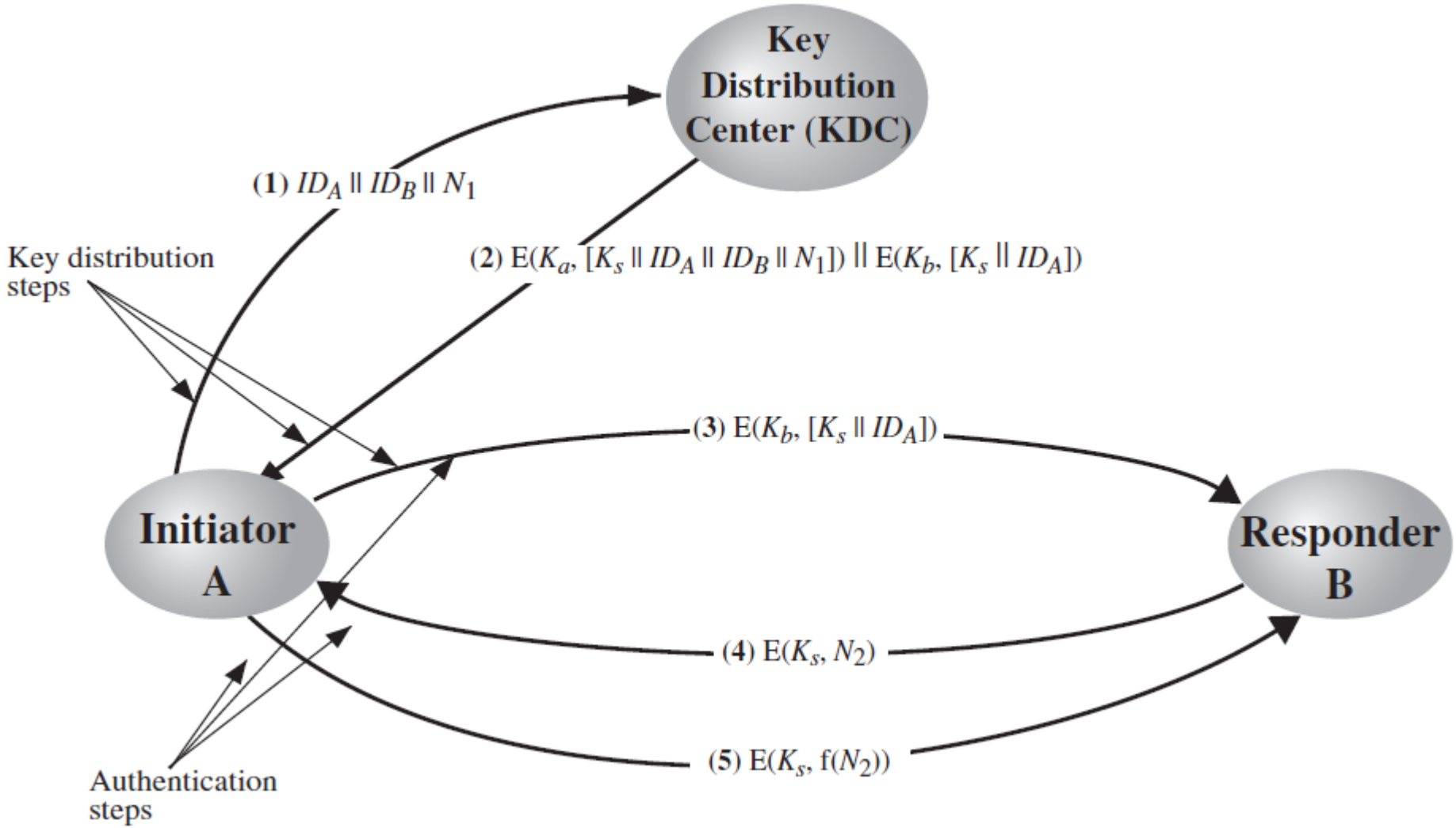
phân phối khóa

K_a, K_b: Khóa chủ

K_s: Khóa phiên

N_1, N_2: Định danh phiên giao dịch

f() là hàm



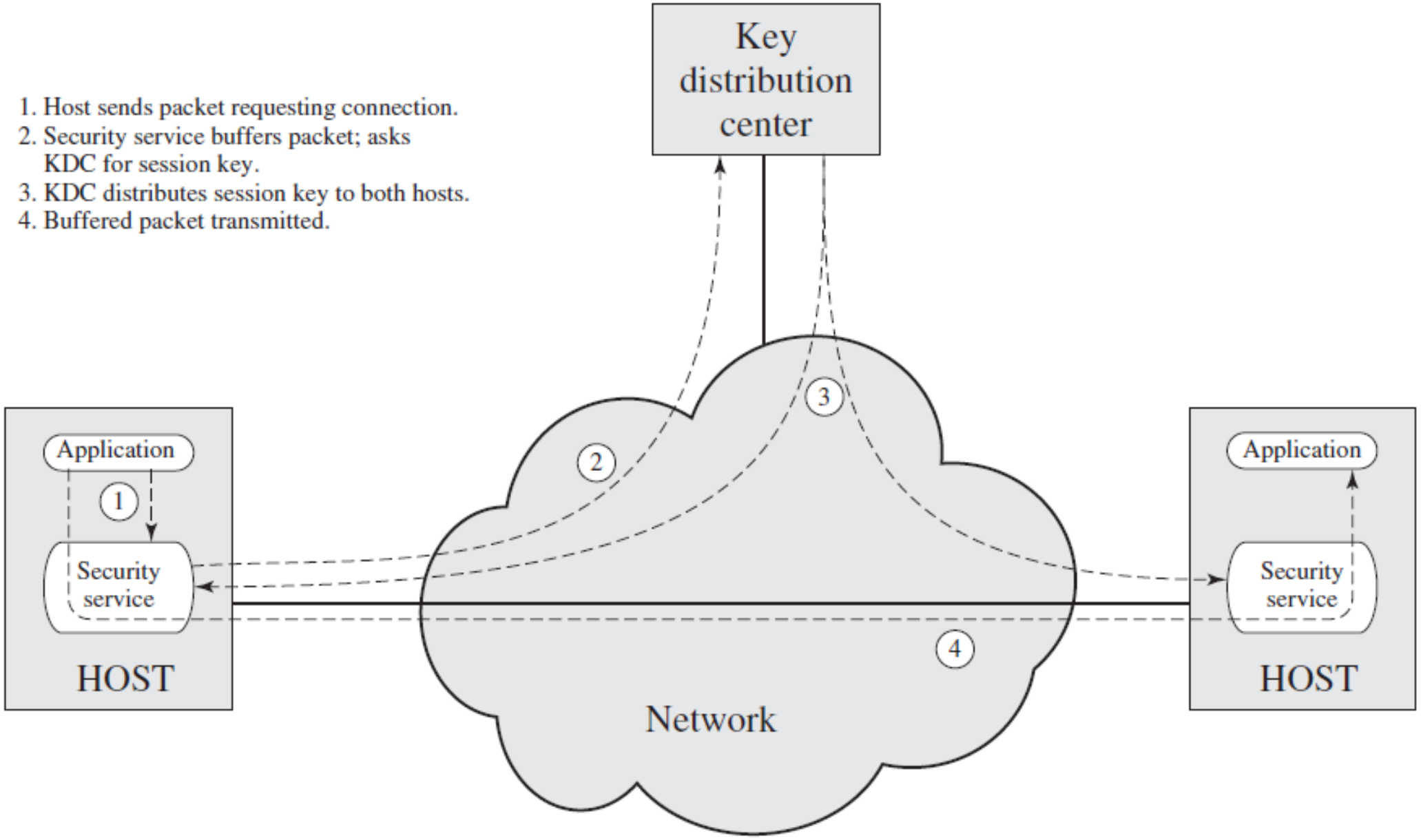
b) Hierarchical Key Control

- Với mạng lớn thì trung tâm phân phối khóa (KDC) có thể bị quá tải. Khi đó người ta dùng KDC phân cấp.
- KDC khu vực dùng cho một miền nhỏ như LAN hay 1 tòa nhà.
- Nếu có 2 thực thể khác miền muốn chia sẻ khóa thì KDC khu vực sẽ kết nối với KDC cấp cao hơn. Khi đó 1 KDC bất kỳ trong 3 KDC sẽ lựa chọn khóa.
- Phân cấp này có thể mở rộng hơn 3 cấp tùy thuộc vào số lượng user và tình trạng địa lý của liên mạng.

c) Thời gian tồn tại của khóa phiên

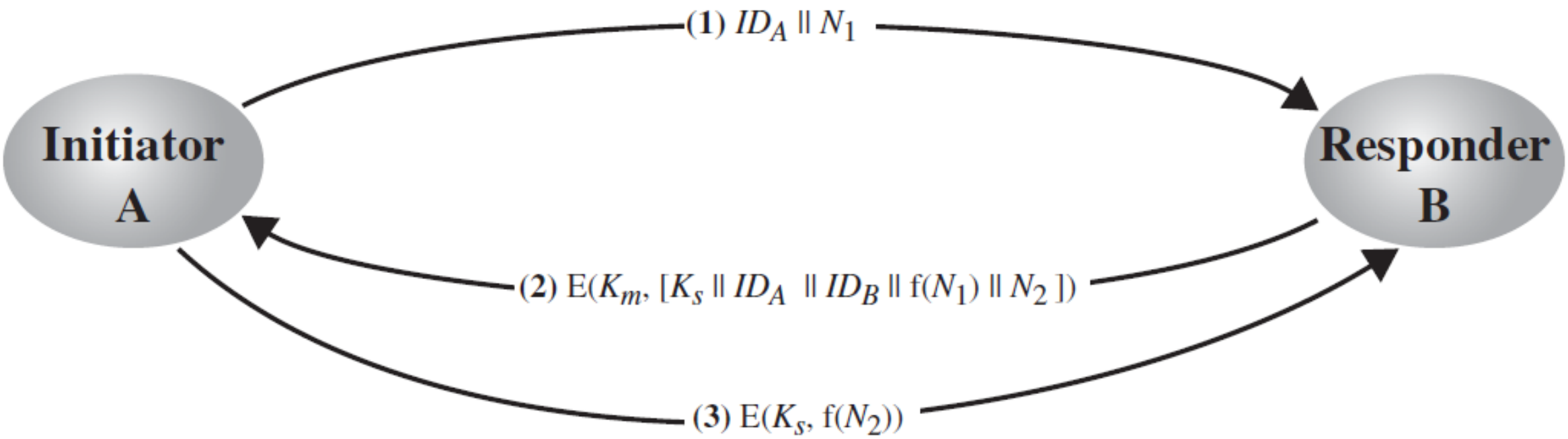
- Tần suất thay đổi khóa phiên càng nhiều thì khóa càng an toàn vì kẻ địch càng ít bản mã cho mỗi khóa phiên. Tuy nhiên phân phối khóa sẽ bị trễ và tạo một gánh nặng lên dung lượng mạng. Người quản trị mạng phải cân bằng các tranh giành này để quyết định thời gian tồn tại của khóa phiên cụ thể.
- Với giao thức hướng kết nối, sử dụng cùng một khóa phiên trong khoảng thời gian kết nối mở. Mỗi phiên kết nối dùng 1 khóa mới. Với phiên kết nối rất lâu thì cần cẩn thận thay đổi khóa theo một chu kỳ, có thể theo chu kỳ của đơn vị dữ liệu giao thức PDU (protocol data unit).
- Đối với giao thức không kết nối thì không có sự rõ ràng bắt đầu và kết thúc kết nối. Sử dụng một khóa phiên cho một số chu kỳ cố định hoặc cho một số giao dịch nhất định.

d) Sơ đồ điều khiển khóa rõ ràng



e) Điều khiển khóa phân tán

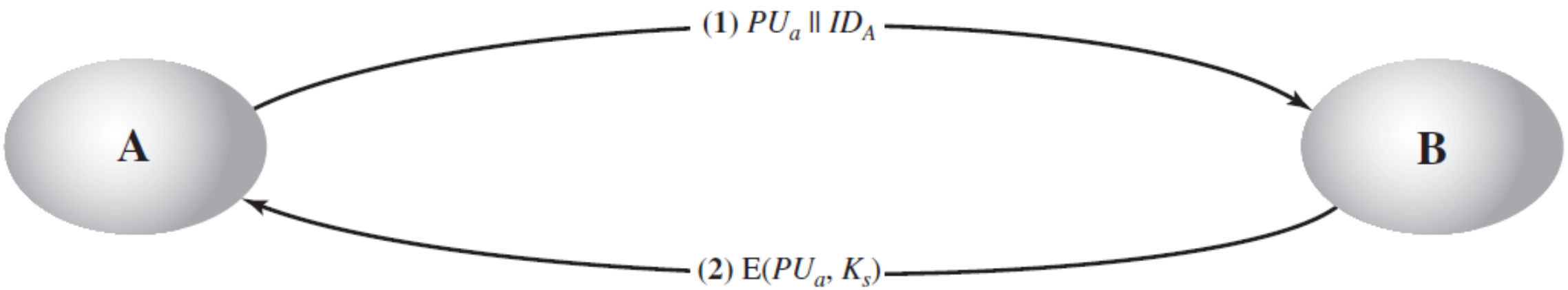
- KDC phải được tin cậy và bảo vệ an toàn. Có thể tránh được điều này bằng cách phân phối phân tán.



Decentralized Key Distribution

2. Phân phối khóa đối xứng dùng mật mã bất đối xứng

a) Phân phối khóa bí mật đơn giản



Simple Use of Public-Key Encryption to Establish a Session Key

2. Phân phối khóa đối xứng dùng mật mã bất đối xứng

a) Phân phối khóa bí mật đơn giản

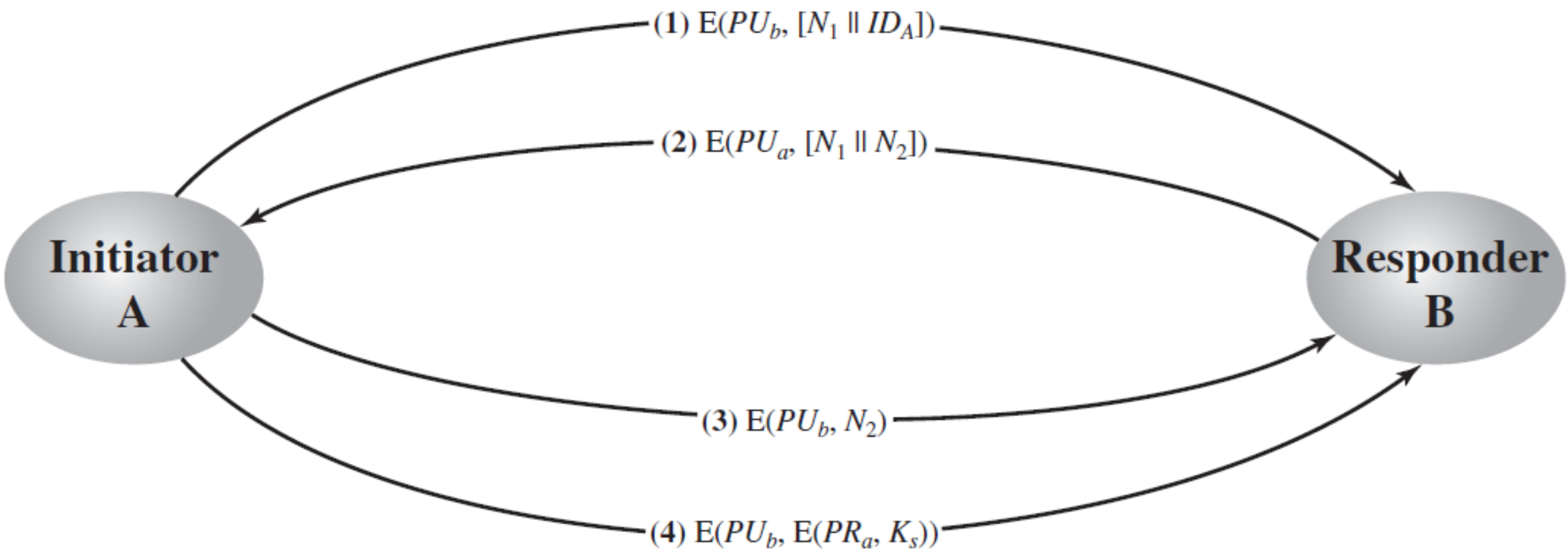
1. A tạo cặp khóa công khai/ bí mật $\{PU_a, PR_a\}$ và truyền tới B thông điệp $PU_a \,||\, ID_A$.
2. B tạo ra khóa bí mật K_s; mã hóa khóa bí mật này cùng với khóa công khai rồi truyền tới A.
3. A giải mã $\mathrm{D}(PR_a, \mathrm{E}(PU_a, K_s))$ để có khóa bí mật K_s.
4. A xóa bỏ khóa công khai, khóa bí mật, B xóa bỏ khóa công khai.

a) Phân phối khóa bí mật đơn giản

- Giao thức này không an toàn nếu như có kẻ địch E tấn công ở vào giữa (man-in-the-middle attack).

1. A tạo cặp khóa công khai/ bí mật $\{PU_a, PR_a\}$ và truyền tới B thông điệp $PU_a \,||\, ID_A$.
2. E can thiệp và tạo cặp khóa công khai/ bí mật $\{PU_e, PR_e\}$ và truyền tới B thông điệp $PU_e \,||\, ID_A$.
3. B tạo ra khóa bí mật Ks và truyền $\mathrm{E}(PU_e, K_s)$. tới A.
4. E can thiệp vào giữa và biết được K_s bằng cách tính $\mathrm{D}(PR_e, \mathrm{E}(PU_e, K_s))$
5. E truyền $\mathrm{E}(PU_a, K_s)$ tới A.

b) Phân phối khóa bí mật có cẩn mật và xác thực



Public-Key Distribution of Secret Keys

c) Sơ đồ lai (Hybrid Scheme)

- Sử dụng KDC để chia sẻ khóa bí mật với mỗi bên và phân phối khóa phiên bí mật được mã hóa với khóa chủ.
- Sơ đồ khóa công khai được sử dụng để phân phối khóa chủ.

3. Phân phối khóa công khai

Có thể nhóm lại thành các sơ đồ sau:

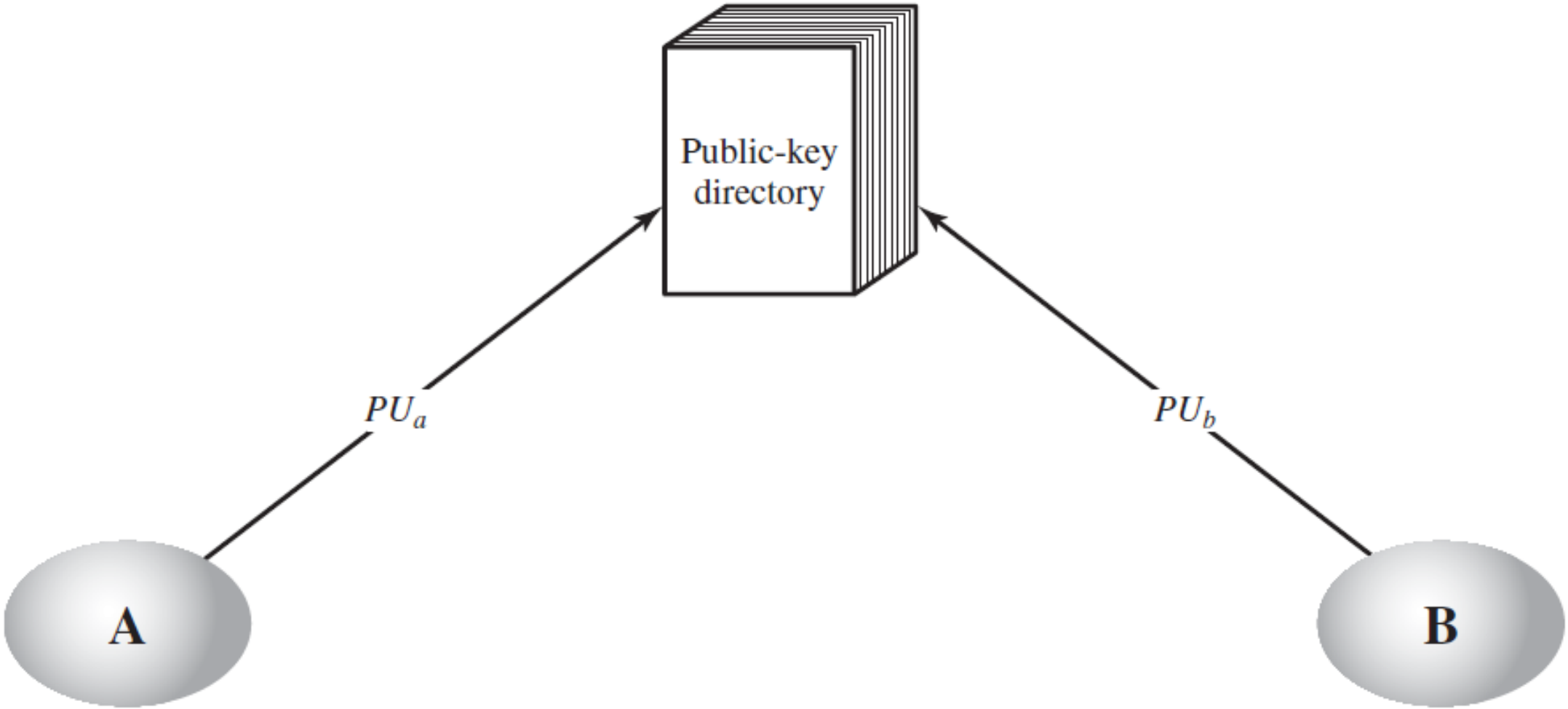
- Thông báo công khai (Public announcement)
- Thư mục sẵn dùng công khai (Publicly available directory)
- Nhà thẩm quyền khóa công khai (Public-key authority)
- Chứng chỉ khóa công khai (Public-key certificates)

a) Thông báo công khai các khóa công khai

- Bất cứ người nào cũng có thể công bố rộng rãi khóa công khai của mình.

- Điểm yếu chính của sơ đồ này là xác thực. Bất cứ người nào cũng có thể giả mạo thông báo của mình.
- Ví dụ, một người nào đó giả mạo là A và thông báo khóa công khai của A. Đến khi A phát hiện ra thì giao dịch bất hợp pháp đã thực hiện rồi.

b) Thư mục sẵn dùng công khai



b) Thư mục sẵn dùng công khai

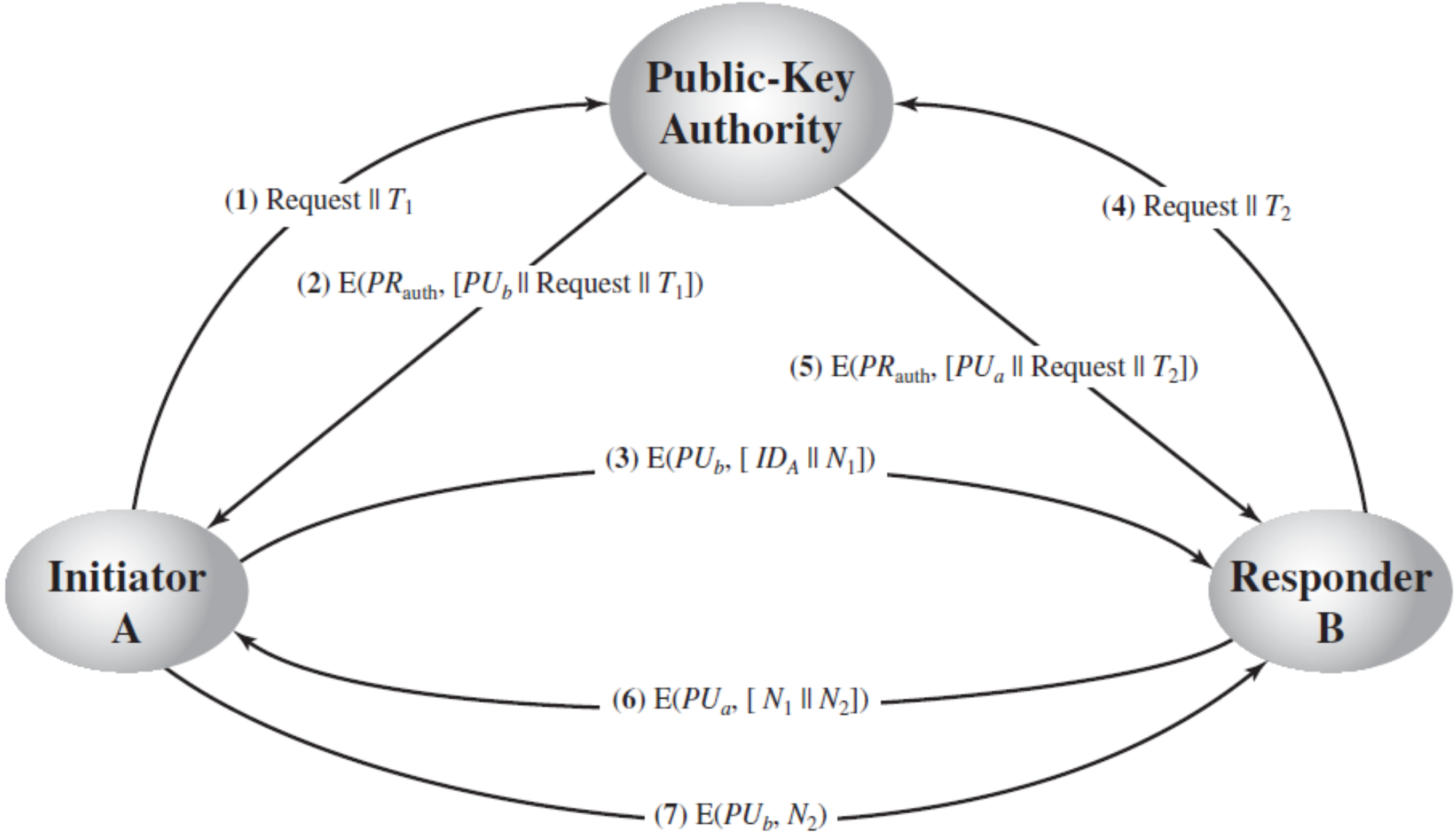
- Trung tâm chứng thực duy trì 1 thư mục cùng với {Tên, Khóa công khai} của mỗi người tham gia
- Mỗi người tham gia đăng ký một khóa công khai. Việc đăng ký phải ở dạng được xác thực bí mật.
- Một người có thể thay đổi khóa của mình bất cư khi nào.
- Các bên tham gia có thể thâm nhập thư mục theo cách điện tử. Việc liên lạc này phải được xác thực và an toàn.

b) Thư mục sẵn dùng công khai

Sơ đồ này vẫn có thể bị tấn công.

- Nếu kẻ địch thành công trong việc nhận được hoặc tính được khóa bí riêng của trung tâm chứng thực thư mục thì nó có thể chuyển khóa công khai giả mạo. Vì vậy các bên tham gia bị nghe lén.
- Kẻ địch cũng có thể lục lọi các bản ghi khóa của trung tâm chứng thực.

c) Nhà thẩm quyền khóa công khai



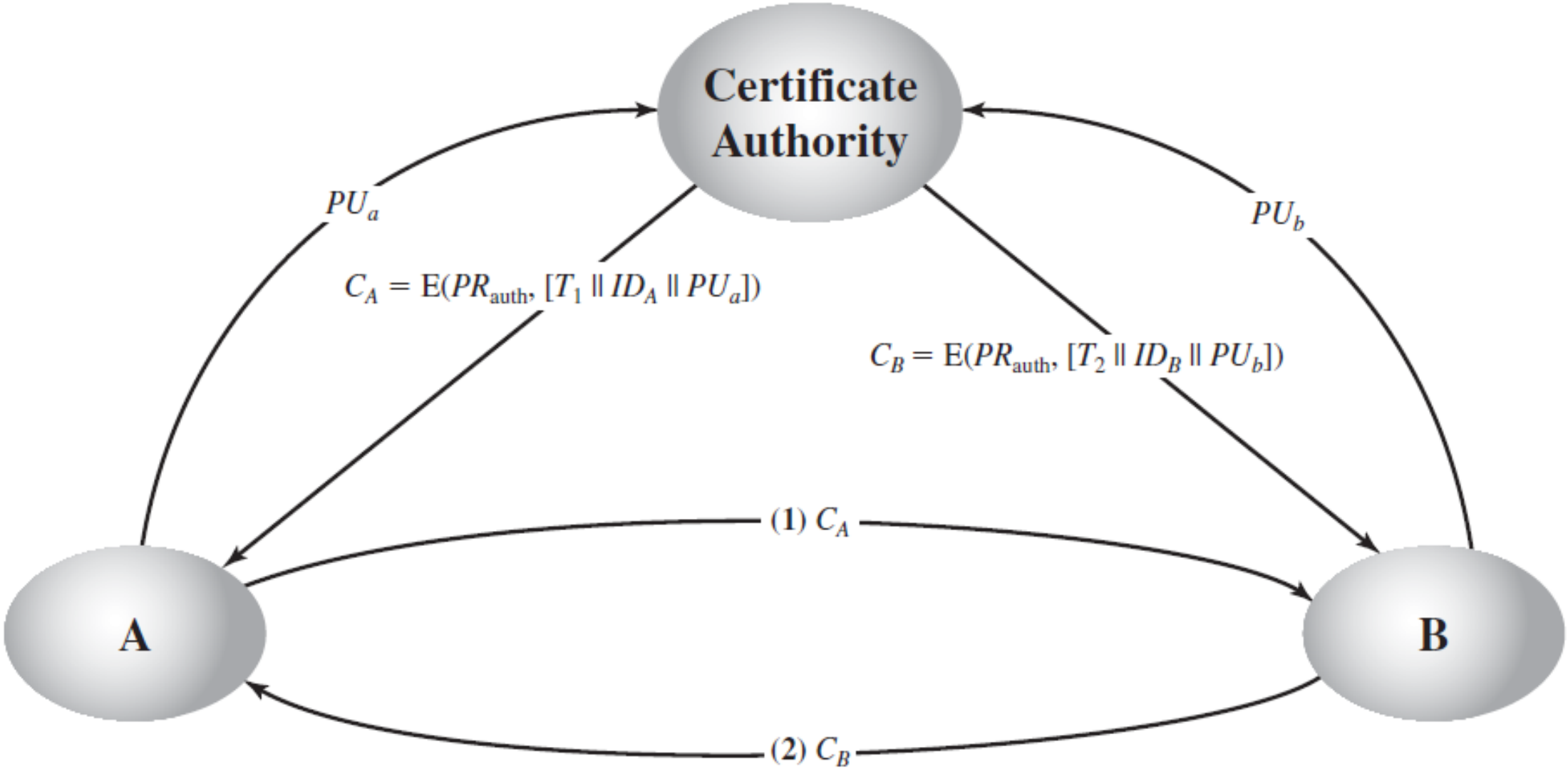
c) Nhà thẩm quyền khóa công khai

- (1) A gửi một thông điệp đánh dấu thời gian tới nhà thẩm quyền yêu cầu khóa công khai của B.
- (2) Nhà thẩm quyền trả lời bằng 1 thông điệp được mã hóa dùng khóa riêng của nhà thẩm quyền PR_{auth} . A có khả năng giải mã sử dụng khóa công khai của nhà thẩm quyền.
- (3) A gửi B thông điệp được mã hóa chứa khóa công khai của B, định danh A và định danh phiên làm việc N_1
- (4, 5) B nhận khóa công khai của A theo cách như A nhận khóa công khai của B
- (6) B gửi thông điệp đã mã hóa tới A chứa PUa và định danh phiên của B (N_2).
- (7) A trả lời bằng thông điệp được mã hóa chứa Pu_b và N_2 để đảm bảo với B rằng đã liên lạc từ A

d) Chứng chỉ khóa công khai

- Sơ đồ (c) ở trên có thể bị tắc nghẽn cổ chai.
- Sử dụng chứng chỉ cho mỗi bên để trao đổi khóa mà không cần kết nối với nhà thẩm quyền khóa công khai.
- Một chứng chỉ bao gồm một khóa công khai, một định danh chủ khóa, và toàn bộ khối được ký bởi bên thứ 3.
- Bên thứ 3 thường là cơ quan cấp chứng chỉ CA (Certificate Authority) như cơ quan thuộc chính phủ hoặc tài chính được tin cậy bởi cộng đồng.
- Một người có thể trình khóa công khai của mình tới CA để nhận được chứng chỉ. Sau đó có thể công khai chứng chỉ.
- Bất cứ người nào cần nhận khóa công khai của người này có thể nhận được chứng chỉ và kiểm tra tính hợp lệ bằng chữ ký tin cậy.

d) Chứng chỉ khóa công khai

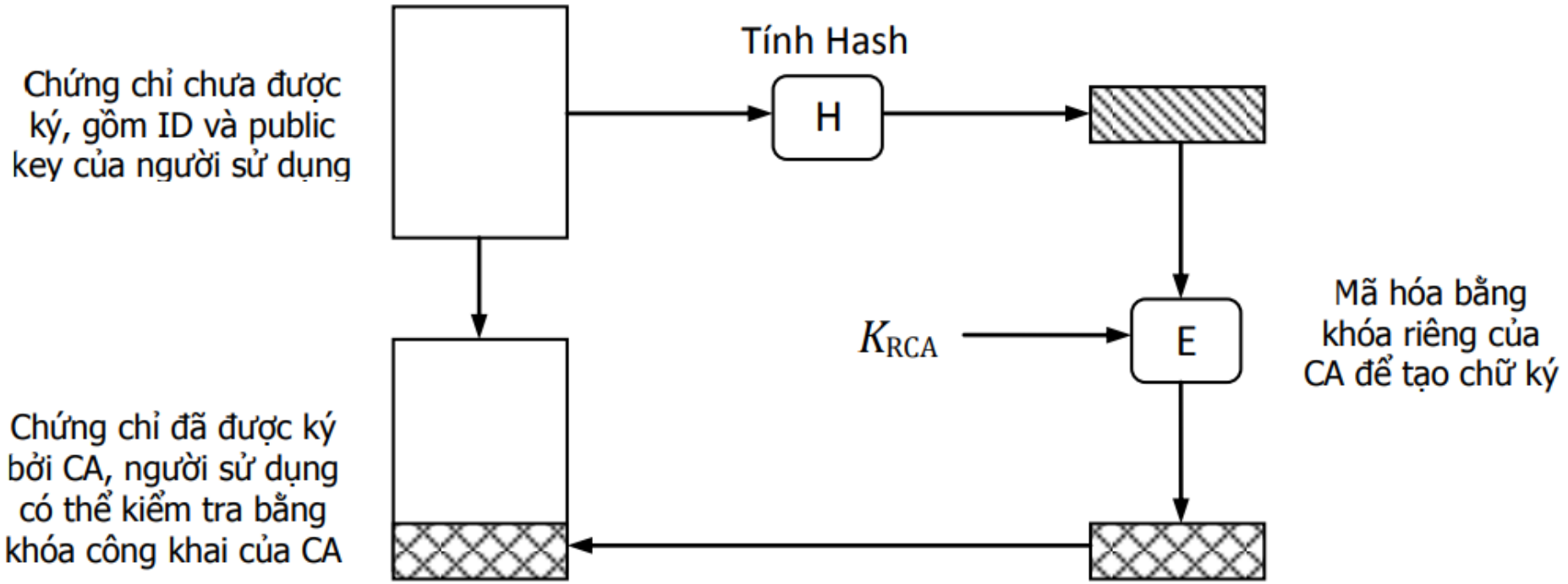


4. Chứng chỉ X.509

- **X.509** ra đời vào năm 1988, và phiên bản mới nhất là v3, là một định dạng tiêu chuẩn cho chứng nhận khoá công khai
- X.509 có nhiệm vụ xác thực danh tính
- X.509 được sử dụng trong hầu hết các ứng dụng an ninh mạng như an ninh IP, an ninh tầng vận chuyển TLS (transport layer security), và S/MINE.

4. Chứng chỉ X.509

Sơ đồ nguyên tắc để sinh ra chứng thực X.509



$$\text{Certificate} = \text{ID}||\text{K}_\text{U}||\text{E}(\text{H}(\text{ID}, \text{K}_\text{U}), \text{K}_\text{RCA})$$

Cấu trúc và ví dụ

Version	Version 3
Serial Number	05:A0:4C
Certificate Signature Algorithm	PKCS #1 SHA-1 With RSA Encryption
Issuer Name	OU = Equifax Secure Certificate Authority; O = Equifax
Validity (Not Before, Not After)	04/01/2006 17:09:06 PM GMT - 04/01/2011 17:09:06 PM GMT
Subject	CN= login.yahoo.com; OU= Yahoo; O= Yahoo! Inc.
Subject Public Key Algorithm	PKCS #1 RSA Encryption
Subject Public Key	30 81 89 02 81 81 00 b5 6c 4f ee ef 1b 04 5d be…
Issuer Unique Identifie	
Subject Unique Identifier	
Extension for version 3	
Certificate Signature Algorithm	PKCS #1 SHA-1 With RSA Encryption
Certificate Signature Value	50 25 65 10 43 e1 74 83 2f 8f 9c 9e dc 74 64 4e…

4. Chứng chỉ X.509

- Version: phiên bản X.509 của chứng chỉ này, có 3 phiên bản là 1, 2 và 3.
- Serial Number: số serial của chứng chỉ này do CA ban hành.
- Certificate Signature Algorithm: thuật toán ký chứng chỉ, gồm loại hàm Hash và phương pháp mã hóa khóa công khai.
- Issuer name: tên của CA (CN: common name, O: organization, OU: organization unit).
- Validity: thời gian hiệu lực của chứng chỉ.
- Subject: tên chủ sở hữu chứng chỉ, cũng gồm có CN, O, OU,…

4. Chứng chỉ X.509

- Subject Public Key Algorithm: thuật toán mã hóa khóa công khai mà tương ứng với khóa công khai trong chứng chỉ.
- Subject Public Key: khóa công khai trong chứng chỉ, tức khóa công khai của chủ sở hữu. Đối với RSA thì thuộc tính này lưu giữ giá trị Modulus và Exponent nối tiếp nhau (N và e).
- Issuer Unique Identifier, Subject Unique Identifier: dành cho version 2, ít được sử dụng.
- Extension: dành cho version 3.
- Certificate Signature Algorithm: thuật toán ký chứng chỉ, giống mục thứ 3.
- Certificate Signature Value: giá trị của chữ ký.

4. Chứng chỉ X.509

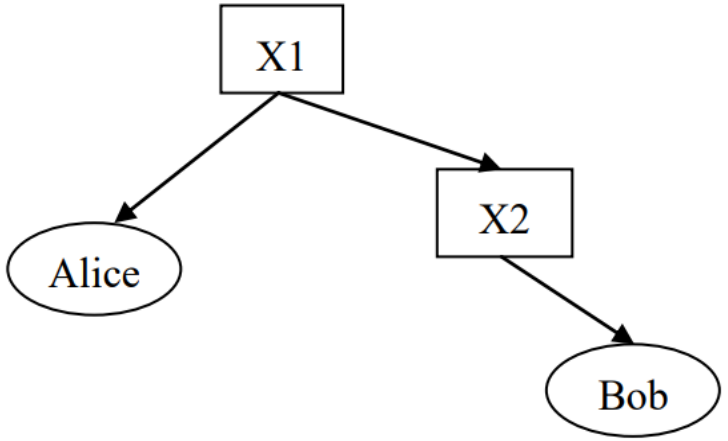
Đối với version 3 phần Extension có thể gồm các thông tin sau:

- Authority key identifier: một con số dùng để định danh của cơ quan cấp chứng chỉ. Thuộc tính Issuer Name cung cấp tên trung tâm chứng thực dưới dạng text, điều này có thể gây nhầm lẫn.
- Subject key identifier: Một con số dùng để định danh người sử dụng được chứng thực. Tương tự như Issuer Name, thuộc tính Subject cũng cung cấp tên người dưới dạng text, điều này có thể gây nhầm lẫn. Ngoài ra việc dùng một con số định danh cho phép một người sử dụng có thể có nhiều chứng chỉ khác nhau.
- Key Usage: mục đích sử dụng của chứng chỉ. Mỗi chứng chỉ có thể có một hoặc nhiều mục đích sử dụng như: mã hóa dữ liệu, mã hóa khóa, chữ ký điện tử, không thoái thác …
- CRL Distribution Point: địa chỉ để lấy danh sách các chứng chỉ đã hết hạn hay bị thu hồi (certificate revocation list).

Phân cấp chứng chỉ

- Trên thế giới không thể chỉ có một cơ quan cấp chứng chỉ CA duy nhất mà có thể có nhiều CA.
- Những người sử dụng khác nhau có thể đăng ký chứng thực tại các CA khác nhau.
- Do đó để có thể trao đổi dữ liệu, một người cần phải tin tưởng vào khóa công khai của tất cả các CA.
- Để giảm bớt gánh nặng này, X.509 đề ra cơ chế phân cấp chứng chỉ.

Phân cấp chứng chỉ



- Alice chỉ tin tưởng vào X1, còn chứng thực của Bob là do X2 cung cấp.
- Nếu Alice không có khóa công khai của X2, thì làm sao Alice có thể kiểm tra được chứng thực của Bob?
- Alice có thể đọc Authority key identifier (tức ID của X2) trong chứng thực của Bob.
- Alice kiểm tra xem X1 có cấp chứng thực nào cho X2 hay không.
- Nếu có, Alice có thể tìm thấy được khóa công khai của X2 và tin tưởng vào khóa này (do đã được X1 xác nhận). Từ đó Alice có thể kiểm tra tính xác thực của chứng chỉ của Bob.
- Có thể có nhiều CA tạo thành một mạng lưới cấp chứng chỉ

Các định dạng file của chứng chỉ X.509

- Dạng DER (.cer): nội dung của chứng chỉ X.509 được lưu dưới format DER, một định dạng dữ liệu binary chuẩn cho các môi trường máy tính.
- Dạng PEM (.pem): là dạng DER và được mã hóa dưới dạng text theo chuẩn Base64. Một file text PEM bắt đầu bằng dòng -----BEGIN CERTIFICATE----- và kết thúc bằng dòng -----END CERTIFICATE-----.
- Dạng PKCS#7 (.p7c hay .p7b): là một định dạng dữ liệu được mã hóa hay ký. Do đó có đi kèm cả chứng chỉ.
- Dạng PKCS#10 (.p10 hay .p10): là một định dạng dùng để gửi yêu cầu cấp chứng chỉ X509 đến trung tâm chứng thực. Định dạng này có ID và public key của người yêu cầu.
- Dạng PKCS#12 (.p12): lưu trữ chứng chỉ X509 và private key tương ứng (có password bảo vệ) trong cùng một file.
- Dạng PFX (.pfx): cũng lưu chứng chỉ X509 và private key theo định dạng của Microsoft.

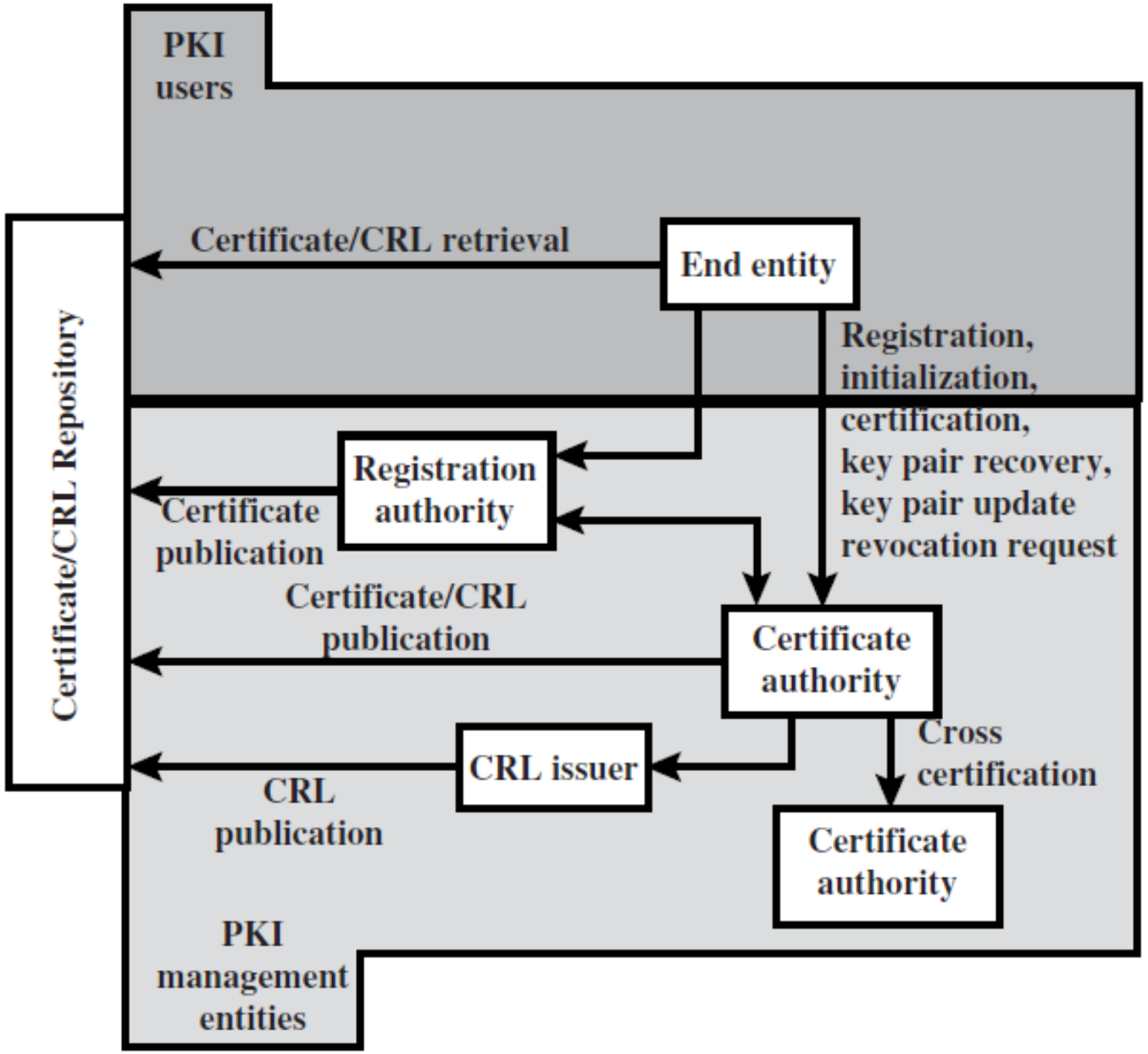
5. Hạ tầng cơ sở khóa công khai

- Hạ tầng cơ sở khóa công khai PKI (public-key infrastructure) là một tập gồm phần cứng, phần mềm, con người, chính sách, và thủ tục cần thiết để tạo, quản lý, phân phối, và kích hoạt chứng chỉ số dựa trên mật mã bất đối xứng.
- Mục đích chính để phát triển PKI là cho phép nhận khóa công khai một cách an toàn, thuận tiện và hiệu quả.
- Hạ tầng cơ sở X.509 (PKIX) dựa trên X.509 là phù hợp để triển khai kiến trúc dựa trên chứng chỉ dùng cho Internet.

5. Hạ tầng cơ sở khóa công khai

Mô hình kiến trúc PKIX

End entity: Là đầu cuối như người dùng, thiết bị (server, router), hay bất kỳ thực thể nào liên quan đến chứng chỉ khóa công khai.



5. Hạ tầng cơ sở khóa công khai

- Certification authority (CA): Cơ quan cấp phát chứng chỉ và danh sách chứng chỉ bị thu hồi CRL (certificate revocation lists). Nó cũng có thể hỗ trợ các chức năng quản trị mặc dù các chức năng này được giao phó cho RA.
- Registration authority (RA): bộ phận đăng ký đảm nhiệm các chức năng từ CA. RA thường liên kết với quá trình đăng ký của thực thể cuối nhưng có thể trợ giúp các mảng khác.
- CRL issuer: Bộ phận mà CA giao cho phát hành CRL.
- Repository: Phương pháp lưu giữ chứng chỉ và CRL để các thực thể đầu cuối có thể nhận lại.

5.2 Xác thực người dùng

1. Nguyên lý xác thực người dùng từ xa

- Trong hầu hết các trường hợp an ninh máy tính, xác thực người dùng là khối cơ bản đầu tiên của phòng vệ.
- Xác thực người dùng là viêc cơ bản cho hầu hết các dạng kiểm soát thâm nhập và cho trách nhiệm giải trình người dùng.
- Xác thực người dùng là khác biệt với xác thực thông điệp.
- Mỗi người dùng đều có một định dạng. Đi kèm với định dạng cá nhân thường là mật khẩu.

1. Nguyên lý xác thực người dùng từ xa

Một quá trình xác thực bao gồm 2 bước:

- Nhận dạng: Trình báo định danh tới hệ thống an ninh.
- Xác minh: Trình báo hoặc tạo thông tin xác thực chứng thực sự ràng buộc giữa thực thể và định danh.

1. Nguyên lý xác thực người dùng từ xa

Có 4 cách để xác thực định danh người dùng:

- Những thứ mà chỉ từng cá nhân biết. Chẳng hạn: mật khẩu, số định danh cá nhân (PIN), hoặc trả lời những câu hỏi được sắp xếp trước.
- Những thứ mà từng cá nhân xử lý: Chẳng hạn: khóa mật mã, thẻ khóa điện tử, thẻ thông minh, khóa vật lý. Dạng này được gọi là vật chứng (token).
- Những dấu hiệu sinh học riêng của cá nhân như vân tay, khuôn mặt, mống mắt.
- Những thứ mà cá nhân làm như giọng nói, chữ viết tay, nhịp điệu gõ phím.

1. Nguyên lý xác thực người dùng từ xa

Xác thực lẫn nhau

- Giao thức này cho phép các bên truyền tin thỏa mãn định danh lẫn nhau và trao đổi khóa phiên.
- Có 2 vấn đề phát sinh: bí mật và đúng lúc.
  - Bí mật: Để tránh giả mạo thì định danh và khóa phiên cần được mã hóa.
  - Đúng lúc: Đe dọa bởi việc lặp lại thông điệp. Việc lặp lại cho phép kẻ địch dàn xếp khóa phiên hoặc đóng giả bên kia.

Xác thực lẫn nhau

Giải quyết vấn đề lặp lại:

- Gửi kèm một dãy số vào thông điệp. Thông điệp chỉ được chấp nhận khi dãy số đúng thứ tự. Khó khăn cho 2 bên khi phải theo dõi dãy số cuối cùng.
- Đánh dấu thời gian vào bản tin. Đòi hỏi 2 bên phải đồng bộ thời gian.
- Thử thách / Đáp ứng: một bên gửi định danh lượt truyền (nonce) và yêu cầu đáp ứng nhận được đúng định danh đó

Xác thực một chiều

- E-mail không yêu cầu người nhận và người gửi online cùng lúc.
- Phong bì thư hoặc phần đầu thư phải rõ ràng sao cho giao thức gửi nhận thư thực hiện đúng. Chẳng hạn trong Simple Mail Transfer Protocol (SMTP) or X.400.
- Thông điệp trong E-mail cũng cần được mã hóa.

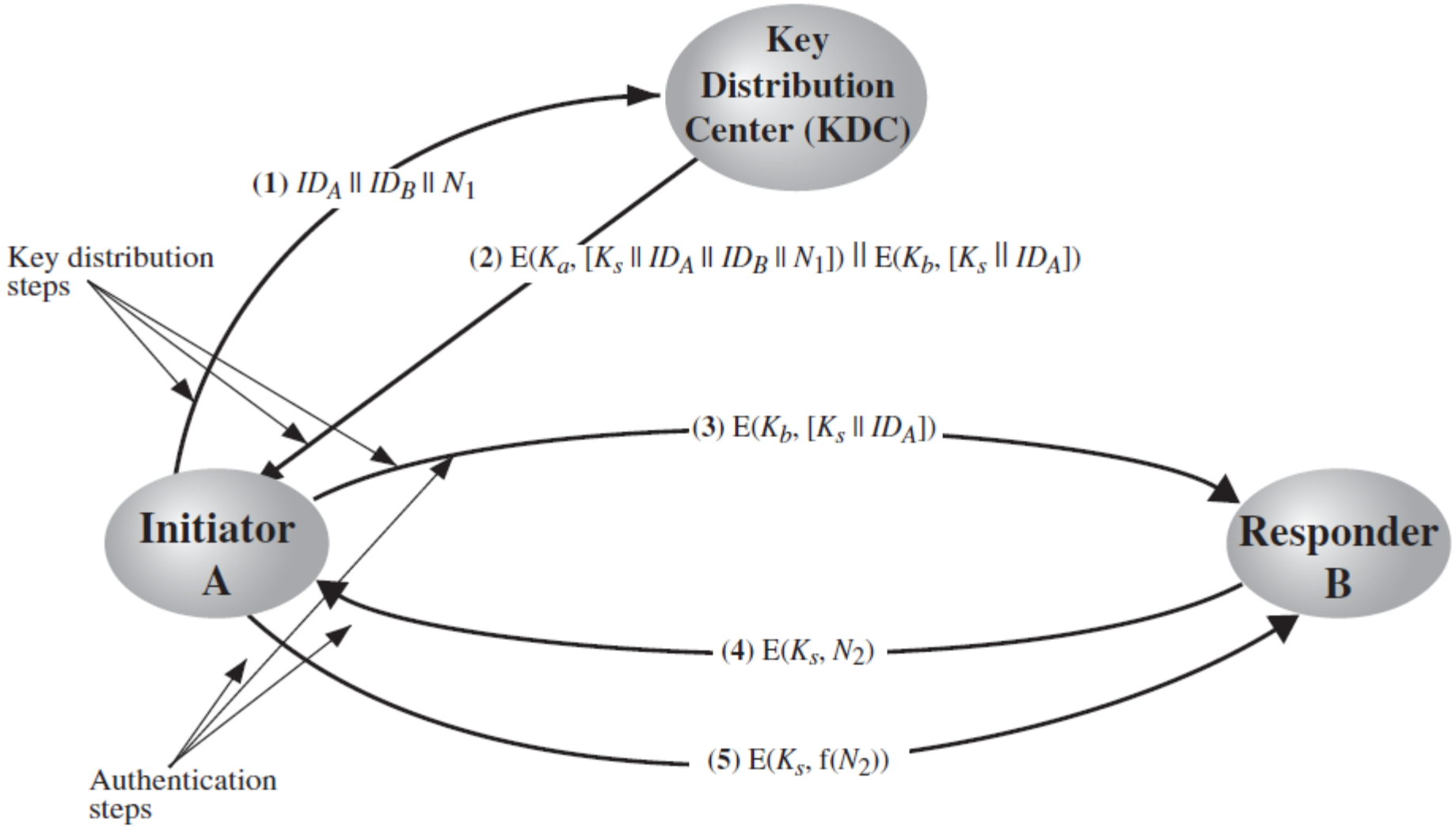
2. Xác thực sử dụng mật mã đối xứng

Xác thực lẫn nhau

Giao thức này

vẫn có thể

bị tấn công



Xác thực lẫn nhau

Giao thức này vẫn có thể bị tấn công

- Kẻ định X có thể đóng giả A và lừa B sử dụng khóa cũ bằng cách chạy lại bước 3.
- Nếu B không nhớ hết tất cả các khóa phiên đã sử dụng với A thì B không thể biết được đây là lặp lại.

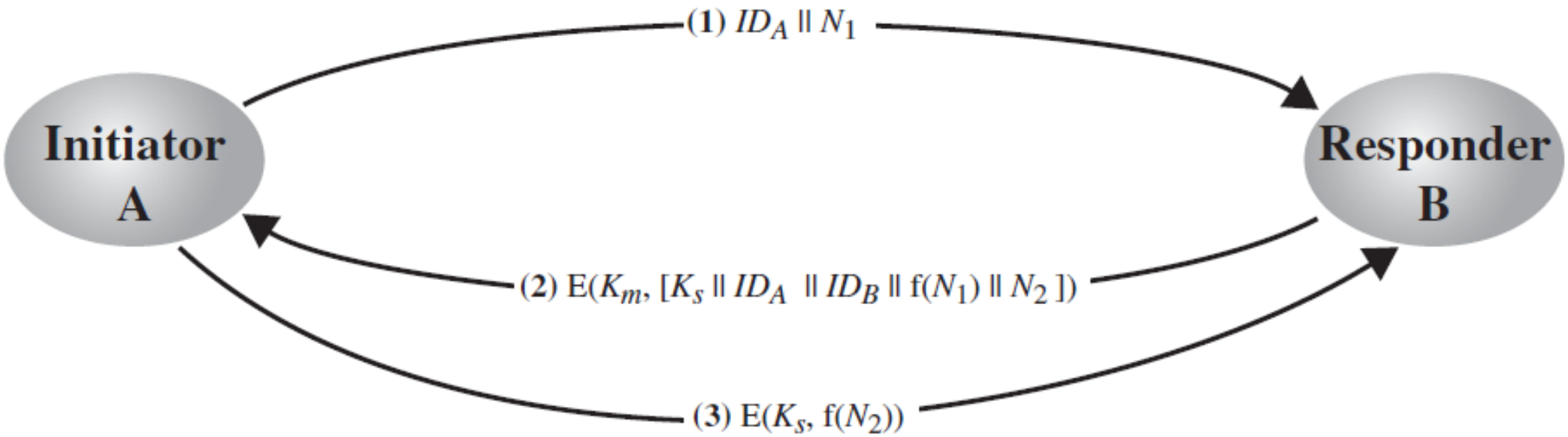
Xác thực lẫn nhau

Khắc phục:

Thêm dấu thời gian T vào bước 2 và 3 các khóa chủ K_a, K_b sẽ an toàn

1. A $\rightarrow$ KDC: $ID_A \| ID_B$
2. KDC $\rightarrow$ A: $E(K_a, [K_s \| ID_B \| T \| E(K_b, [K_s \| ID_A \| T])])$
3. A $\rightarrow$ B: $E(K_b, [K_s \| ID_A \| T])$
4. B $\rightarrow$ A: $E(K_s, N_1)$
5. A $\rightarrow$ B: $E(K_s, f(N_1))$

Xác thực một chiều



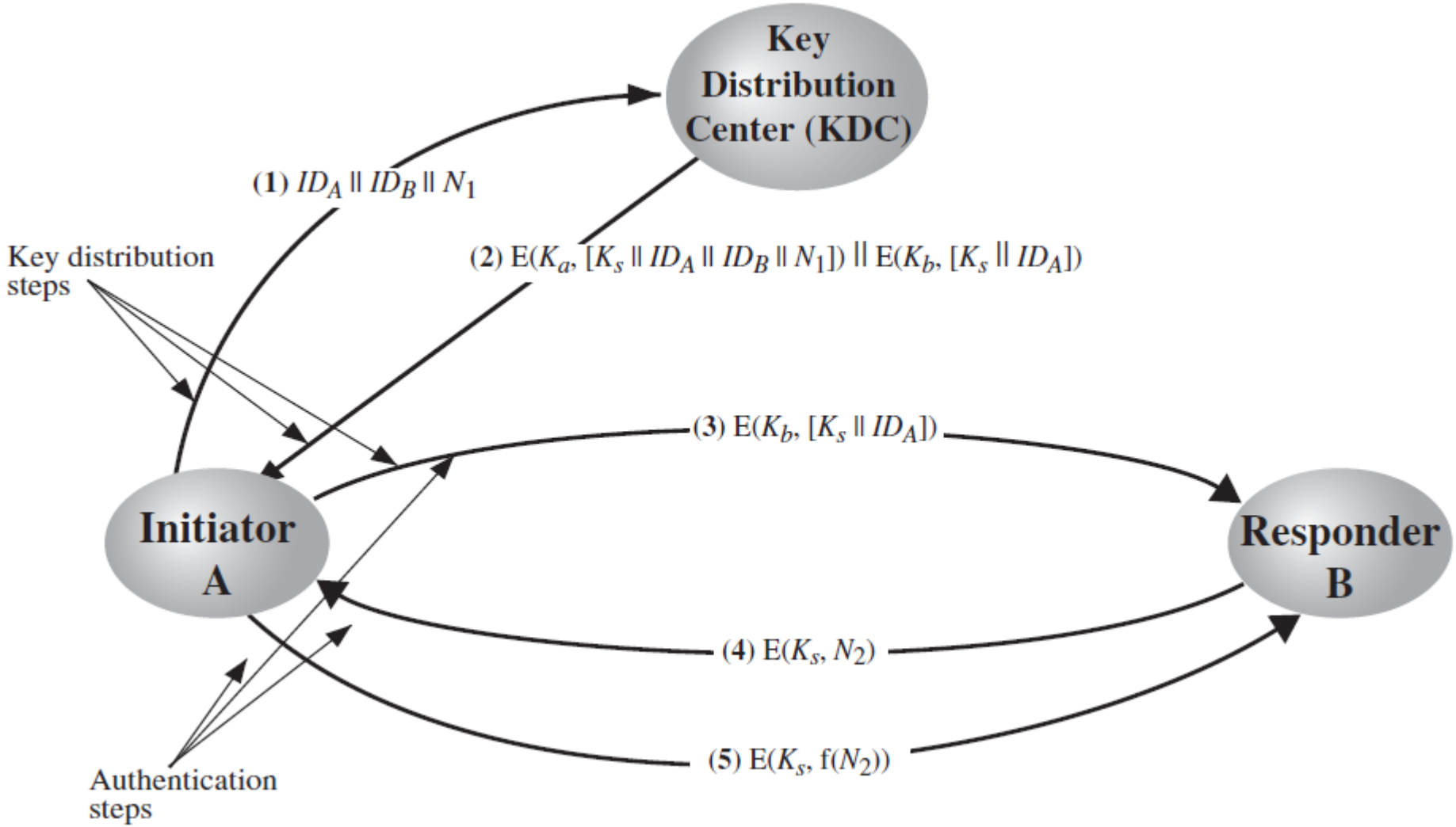
Decentralized Key Distribution

- Phân phối khóa phân tán sử dụng mã hóa đối xứng là không thực tế cho xác thực một chiều
- Người A phải chờ đợi trả lời khóa phiên từ người B thì mới được gửi thông điệp.

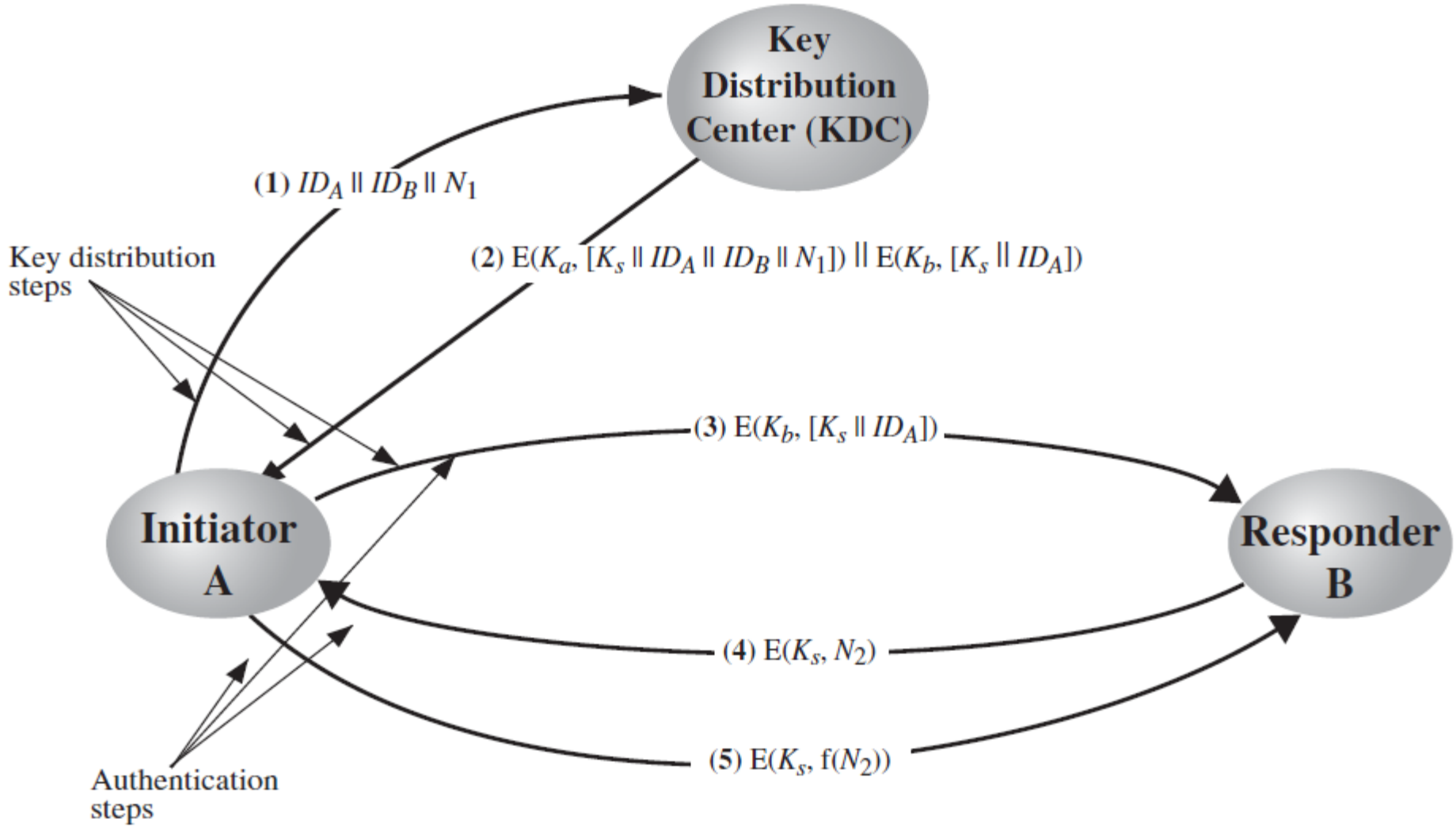
Xác thực một chiều

Chỉ cần chỉnh sửa sơ đồ này cho Email.

Bước 4, 5 bị loại bỏ



1. A → KDC: $ID_A \| ID_B \| N_1$
2. KDC → A: $E(K_a, [K_s \| ID_B \| N_1 \| E(K_b, [K_s \| ID_A])])$
3. A → B: $E(K_b, [K_s \| ID_A]) \| E(K_s, M)$



3. Kerberos

- Mô hình Hệ thống khoá máy chủ tin cậy của MIT
- Cung cấp xác thực có bên thứ ba dùng khoá riêng và tập trung.
- Cho phép người sử dụng truy cập vào các dịch vụ phân tán trong mạng.
- Không cần thiết phải tin cậy mọi máy trạm, thay vì đó chỉ cần tin cậy máy chủ xác thực trung tâm.
- Đã có hai phiên bản đang sử dụng là: Kerberos 4 và Kerberos 5.

Các yêu cầu của Kerberos

- An toàn: Kẻ nghe lén không thể nhận được thông tin cần thiết để đóng giả người dùng.
- Tin cậy: Sử dụng kiến trúc máy chủ phân tán cùng với hệ thống sao lưu.
- Trong suốt: Ngoài việc đăng nhập bằng mật khẩu, người dùng không cần để ý đến việc xác thực.
- Có thể mở rộng: Hệ thống có khả năng hỗ trợ số lượng lớn các máy chủ và máy khách.

Kerberos 4

- Dùng bên thứ ba làm máy chủ xác thực AS (authentication server).
- Người dùng thỏa thuận AS về danh tính của mình
- AS cung cấp sự tin cậy xác thực thông qua thẻ TGT (ticket-granting ticket) máy chủ cấp phát thẻ TGS (ticket-granting server).
- Người sử dụng thường xuyên yêu cầu TGS cho truy cập đến các dịch vụ khác dựa trên thẻ TGT của người sử dụng.

Kerberos 4

- Người sử dụng nhận thẻ được cấp từ máy AS, mỗi thẻ cho một phiên làm việc và cũng nhận thẻ cấp dùng dịch vụ từ TGT.
- Mỗi thẻ dùng cho một dịch vụ khác nhau được yêu cầu, thông qua việc trao đổi giữa máy chủ/trạm để nhận được dịch vụ.
- Lãnh địa Kerberos: máy chủ Kerberos, một số máy trạm đã được đăng ký với máy chủ, các máy chủ ứng dụng chia sẻ khoá với máy chủ.

Kerberos 5

- Được phát triển vào giữa những năm 1990, được thiết kế theo chuẩn RFC 1510.
- Nó cung cấp những cải tiến so với phiên bản 4, cụ thể hướng tới các thiếu xót về môi trường, thuật toán mã, thủ tục mạng thứ tự byte, thời gian sử dụng thẻ, truyền tiếp xác thực, xác thực lãnh địa con.
- Các sự khác biệt về kỹ thuật như: mã kép, các dạng sử dụng không chuẩn, khoá phiên, chống tấn công mật khẩu.

Mô tả giao thức Kerberos

• Trung tâm phân phối khóa bao gồm hai phần riêng biệt: một máy chủ chứng thực và một máy chủ cấp thẻ.

• Làm việc dựa trên các thẻ để thực hiện quá trình chứng thực người dùng.

• Duy trì một cơ sở dữ liệu chứa các khoá bí mật.

• Mỗi thực thể trên mạng (máy trạm hoặc máy chủ) đều chia sẽ một khoá bí mật chỉ giữa bản thân nó với Kerberos.

• Để thực hiện quá trình giao tiếp giữa hai thực thể, Kerberos tạo ra một khoá phiên. Khóa này dùng để bảo mật quá trình tương tác giữa các thực thể với nhau.

Hoạt động của Kerberos

- Người dùng nhập vào tên truy cập và mật khẩu ở phía máy trạm.
- Máy trạm thực hiện thuật toán băm một chiều trên mật khẩu được nhập vào và nó trở thành khoá bí mật của máy trạm.
- Máy trạm gửi một thông điệp dưới dạng bản rõ đến AS để yêu cầu dịch vụ. Không có khoá bí mật cũng như mật khẩu nào được gởi đến AS.

Hoạt động của Kerberos

- AS kiểm tra xem có tồn tại người dùng C trong cơ sở dữ liệu của nó hay không. Nếu có, nó gởi ngược lại cho máy trạm hai thông điệp:
  - Thông điệp A: chứa khoá phiên Máy trạm/TGS được mã hóa bởi khoá bí mật của người dùng.
  - Thông điệp B: chứa Thẻ (bao gồm ID của máy trạm, địa chỉ mạng của máy trạm, kỳ hạn thẻ có giá trị và một khoá phiên máy trạm/TGS) được mã hóa sử dụng khoá bí mật của TGS.
- Khi máy trạm nhận được thông điệp A và B, nó giải mã thông điệp A để lấy khoá phiên máy trạm/TGS. Khoá phiên này được sử dụng cho quá trình giao đổi tiếp theo với TGS. Ở đây máy trạm không thể giải mã thông điệp B bởi vì nó được mã hóa bởi khoá bí mật của TGS.

Hoạt động của Kerberos

- Khi yêu cầu dịch vụ (S), máy trạm gửi hai thông điệp sau đến TGS:
  - Thông điệp C: Gồm thông điệp B và ID của dịch vụ được yêu cầu
  - Thông điệp D: chứa Authenticator (gồm ID máy trạm và nhãn thời gian -timestamp) được mã hóa bởi khoá phiên Máy trạm/TGS.
- Khi nhận được thông điệp C và D, TGS giải mã thông điệp D sử dụng khoá phiên máy trạm/TGS và gửi hai thông điệp ngược lại cho máy trạm:
  - Thông điệp E: chứa thẻ (máy trạm đến máy chủ) (bao gồm ID máy trạm, địa chỉ mạng của máy trạm, kỳ hạn thẻ có giá trị và một khoá phiên máy trạm/dịch vụ) được mã hóa bởi khoá bí mật của dịch vụ.
  - Thông điệp F: chứa khoá phiên của máy trạm/máy chủ được mã hóa bởi khoá phiên máy trạm/TGS.

Hoạt động của Kerberos

- Khi nhận được thông điệp E và F, máy trạm sau đó gởi một Authenticator mới và một thẻ (máy trạm đến máy chủ) đến máy chủ chứa dịch vụ được yêu cầu.
  - Thông điệp G: chứa thẻ (máy trạm đến máy chủ) được mã hóa sử dụng khoá bí mật của máy chủ.
  - Thông điệp H: một Authenticator mới chứa ID máy trạm, Timestamp và được mã hóa sử dụng khoá phiên máy trạm/máy chủ.

Hoạt động của Kerberos

- Sau đó, máy chủ giải mã thẻ sử dụng khoá bí mật của chính nó, và gởi một thông điệp cho máy trạm để xác nhận tính hợp lệ thực sự của máy trạm và sự sẵn sàng cung cấp dịch vụ cho máy trạm.
  - Thông điệp I: chứa giá trị Timestamp trong Authenticator được gởi bởi máy trạm sẽ được cộng thêm 1, được mã hóa bởi khoá phiên máy trạm/máy chủ.
- Máy trạm sẽ giải mã sự xác nhận này sử dụng khóa chia sẽ giữa nó với máy chủ, và kiểm tra xem giá trị timestamp có được cập nhật đúng hay không. Nếu đúng, máy trạm có thể tin tưởng máy chủ và bắt đầu đưa ra các yêu cầu dịch vụ gởi đến máy chủ.
- Máy chủ cung cấp dịch vụ được yêu cầu đến máy trạm.

Hạn chế của Kerberos

- Không thật thích hợp cho một số chức năng như ký điện tử (yêu cầu đáp ứng cả hai nhu cầu xác thực và bảo đảm không chối cãi được).
- Một trong những giả thiết quan trọng của giao thức Kerberos là các máy chủ trên mạng cần phải tin cậy được.
- Ngoài ra, nếu người dùng chọn những mật khẩu dễ đoán thì hệ thống dễ bị mất an toàn trước kiểu tấn công từ điển, tức là kẻ tấn công sẽ sử dụng phương thức đơn giản là thử nhiều mật khẩu khác nhau cho đến khi tìm được giá trị đúng.
- Do hệ thống hoàn toàn dựa trên mật khẩu để xác thực người dùng, nếu bản thân các mật khẩu bị đánh cắp thì khả năng tấn công hệ thống là không có giới hạn. Điều này dẫn đến một yêu cầu rất căn bản là Trung tâm phân phối khóa cần được bảo vệ nghiêm ngặt. Nếu không thì toàn bộ hệ thống sẽ trở nên mất an toàn.

4. Xác thực sử dụng mật mã bất đối xứng

Xác thực lẫn nhau

1. $A \rightarrow AS: \quad ID_A \| ID_B$
2. $AS \rightarrow A: \quad E(PR_{as}, [ID_A \| PU_a \| T]) \| E(PR_{as}, [ID_B \| PU_b \| T])$
3. $A \rightarrow B: \quad E(PR_{as}, [ID_A \| PU_a \| T]) \| E(PR_{as}, [ID_B \| PU_b \| T]) \| E(PU_b, E(PR_a, [K_s \| T]))$

- Khóa phiên được chọn và mã hóa bởi A nên không bị nguy hiểm bởi AS
- Dấu thời gian chống lại việc lặp lại
- Giao thức này yêu cầu đồng hồ đồng bộ

Xác thực lẫn nhau

1. A → KDC: $ID_A \| ID_B$
2. KDC → A: $E(PR_{auth}, [ID_B \| PU_b])$
3. A → B: $E(PU_b, [N_a \| ID_A])$
4. B → KDC: $ID_A \| ID_B \| E(PU_{auth}, N_a)$
5. KDC → B: $E(PR_{auth}, [ID_A \| PU_a]) \| E(PU_b, E(PR_{auth}, [N_a \| K_s \| ID_B]))$
6. B → A: $E(PU_a, [E(PR_{auth}, [(N_a \| K_s \| ID_B)]) \| N_b])$
7. A → B: $E(K_s, N_b)$

Giao thức này sử dụng định danh lần truyền N_a, N_b.

Vẫn bị tấn công

Xác thực lẫn nhau

1. A → KDC: $ID_A \| ID_B$
2. KDC → A: $E(PR_{auth}, [ID_B \| PU_b])$
3. A → B: $E(PU_b, [N_a \| ID_A])$
4. B → KDC: $ID_A \| ID_B \| E(PU_{auth}, N_a)$
5. KDC → B: $E(PR_{auth}, [ID_A \| PU_a]) \| E(PU_b, E(PR_{auth}, [N_a \| K_s \| ID_A \| ID_B]))$
6. B → A: $E(PU_a, [E(PR_{auth}, [(N_a \| K_s \| ID_A \| ID_B) \| N_b]))$
7. A → B: $E(K_s, N_b)$

- ID_A được thêm vào bản mã cùng với khóa riêng của *KDC* ở bước 5, 6.
- Ràng buộc khóa riêng K_s tới 2 bên.

Xác thực một chiều

- Khi A gửi Email cho B $\quad A \rightarrow B: \quad E(PU_b, K_s) \| E(K_s, M)$
- Nếu cần xác thực thì cần dùng hàm băm $\quad A \rightarrow B: \quad M \| E(PR_a, H(M))$
- A không thể chối bỏ việc gửi tin. Tuy nhiên kỹ thuật này mở ra một dạng gian lận khác:
- Bob viết Email có ký tên gửi cho giám đốc Alice nêu một ý tưởng hay. Giả sử Max nghe được ý tưởng của Bob và vào được hàng đợi trước khi thư phân phát. Thì Max sẽ tìm được bản tin của Bob. Max cắt chữ ký Bob, thêm vào chữ ký Max và gửi tới Alice. Như vậy Max đã dành lấy ý tưởng của Bob.

Xác thực một chiều

Giải quyết: $A \rightarrow B: \quad E(PU_b, [M \| E(PR_a, H(M))])$

- Cả bản tin và chữ ký đều phải mã hóa với chữ ký công khai của người nhận
- Cần yêu cầu khóa công khai cả A và B cùng với dấu thời gian cùng với khóa riêng của máy chủ xác thực:

$$A \rightarrow B: \quad M \| E(PR_a, H(M)) \| E(PR_{as}, [T \| ID_A \| PU_a])$$

5. Xác thực mật khẩu

- Dựa trên điều mà thực thể biết: người sử dụng đưa ra một mật khẩu và hệ thống sẽ xác minh nó.
- Nếu mật khẩu quả thật là cái được đăng ký trước với người sử dụng, danh tính của người sử dụng sẽ được xác thực. Ngược lại, mật khẩu sẽ bị từ chối và thủ tục xác thực thất bại.
- Thông thường mật khẩu là một chuỗi ký tự có độ dài xác định; ký tự mật khẩu phải được chọn từ một bộ (bảng) ký tự qui định trước. Không gian mật khẩu là tập tất cả các mật khẩu có thể xây dựng được từ qui ước mật khẩu.

5. Xác thực mật khẩu

- Để đảm bảo an toàn, người ta không lưu trữ mật khẩu ở dạng bản rõ tại máy chủ. Vì bản rõ bị lộ sẽ dễ dàng sử dụng.
- OS luôn xây dựng A (tập mật khẩu) và C (tập thông tin đối chiếu lưu trữ phía hệ thống) là khác nhau.
- Các hàm $f \in F$ được sử dụng để biến đổi một giá trị $a \in A$ về $c=f(a) \in C$ để đối chiếu.
- Hàm f thường dùng là hàm băm

5. Xác thực mật khẩu

Tấn công mật khẩu:

- Nếu một người sử dụng có thể đoán được mật khẩu của người khác thì kẻ đó có thể mạo danh người này
- Mục đích của kẻ tấn công chính là đề tìm một giá trị $a \in A$ sao cho với một $f \in F$ nào đó, sẽ có $f(a) = c \in C$; c chính là thành phần đối chiếu ứng với thực thể bị tấn công.
- Việc đoán mật khẩu của một người sử dụng nào đó thành công cần thông qua việc xác định xem một mật khẩu a (đoán) có gắn liền với một người sử dụng đó hay không, tức là thông qua việc thực hiện f(a) hay xác thực bằng thủ tục l(a).

Tấn công mật khẩu

Hai tiếp cận để bảo vệ mật khẩu, được sử dụng đồng thời:

- Che dấu đủ thông tin để một trong các thành phần a, c hay f là không thể tìm thấy.
- Chống truy nhập đến các hàm xác thực trong L. Từ đó chúng ta thấy sẽ có nhiều kiểu tấn công cũng như cơ chế bảo vệ khác nhau.

Tấn công mật khẩu

Tấn công từ điển:

- Thử vét cạn một tập mật khẩu khả nghi thiết lập sẵn (từ điển).
- Đoán mật khẩu dựa vào một số thông tin như các dạng/kết cấu mật khẩu hay được sử dụng và các thông tin cá nhân liên quan có thể có được như tên, tuổi, ngày sinh, số điện thoại, tên người thân cận …

Tấn công mật khẩu

Thử vét cạn từ điển có thể tiến hành theo 2 cách:

- Tấn công ngoại tuyến (off-line attack): đòi hỏi kẻ tấn công phải truy cập được tới tập thông tin đối chứng (tập C) và biết các hàm xác minh. Từ đó kẻ địch chỉ việc tiến hành thử lần lượt mỗi mật khẩu trong từ điển, xem giá trị thu được khi tác động bằng một hàm xác minh có rơi vào tập C hay không.
- Tấn công trực tuyến (on-line attack): đòi hỏi kẻ tấn công phải truy nhập (gọi tới) được các hàm logic L, để lần lượt gọi kiểm tra xem l(g) có trả lại thành công, với mỗi mật khẩu g trong từ điển, và hàm l từ L. Ví dụ: đoán-thử bằng cách gọi chức năng login vào hệ thống.

Các cơ chế phòng vệ

- Phòng vệ qua cơ chế mật khẩu
- Cơ chế làm chậm tấn công từ điển

Phòng vệ qua cơ chế mật khẩu

- Mật khẩu cần được tạo ra sao cho khó đoán.
- Lý tưởng là sinh mật khẩu ngẫu nhiên, tức là đảm bảo xác suất chọn mỗi mật khẩu trong không gian cho phép là như nhau.
- Tuy nhiên mật khẩu ngẫu nhiên là quá khó nhớ nên thường không được dùng.

Phòng vệ qua cơ chế mật khẩu

- Dựa vào các thông tin cá nhân, ví dụ như tên tài khoản, tên người dùng, tên máy tính hoặc địa điểm, mã số thẻ các loại, số điện thoại, ngày sinh …
- Một số người dùng cũng chọn và ghép các từ trong từ điển (các loại, các ngôn ngữ khác nhau).
- "proactive password checking", tức là mật khẩu đã chọn của người sử dụng sẽ được hệ thống kiểm tra đánh giá trước.

Cơ chế làm chậm tấn công từ điển

- Cơ chế này thường gọi là thêm muối
- Hệ thống "trộn thêm" một chuỗi bit ngẫu nhiên vào chuỗi mật khẩu cung cấp của người dùng khi đăng nhập, trước khi tiến hành thủ tục băm và chuyển cho các thao tác kiểm tra tiếp theo.
- Không gian mật khẩu coi như được nở ra theo hàm mũ nhờ vào việc trộn chuỗi bit ngẫu nhiên (hay gọi là các bit muối – salt bit).
- Chuỗi bit này có thể coi là một tham số khóa của hệ thống và được hệ thống lưu trữ theo tên người dùng.
- Kẻ tấn công hoàn toàn không thể đoán được chuỗi bit này (ngẫu nhiên), nên bắt buộc phải thử tấn cả các khả năng của nó, dù chỉ là thử một mật khẩu đoán thử nào đó. Vì vậy quá trình tấn công sẽ bị làm chậm 2^k lần, với k là độ dài chuỗi bit muối.

Cơ chế làm chậm tấn công từ điển

Cách thu ngắn số lần thử mật khẩu:

- Có thể tăng thời gian trễ giữa hai lần thử không thành công theo một hàm tăng nhanh, ví dụ hàm mũ.
- Có thể đặt ngưỡng cho phép gõ sai mật khẩu và bắt dừng khá lâu khi bị vượt ngưỡng, thậm chí tháo bỏ quyền đăng nhập.
- Có thể giảm lỏng, tức là đưa vào một môi trường mô phỏng thử nghiệm để nghiên cứu hành vi của kẻ tấn công.
- Qui định chu kỳ người sử dụng phải thay đổi mật khẩu.