Chương 2:
Mã hóa và Các giao thức trao đổi khóa

Khoa Khoa học và Kỹ thuật Máy tính
Đại học Bách Khoa Tp.HCM

Nội dung

1. Những khái niệm cơ bản về mã hóa
2. Mã hóa hoàn hảo
3. Kênh trao đổi khóa
4. Mô hình Dolev-Yao
5. Giao thức trao đổi khóa

Tài liệu tham khảo:

W. Mao (2003). ***Modern Cryptography: Theory and Practice***, 3rd Ed., Prentice Hall, ISBN 0-13-066943-1.

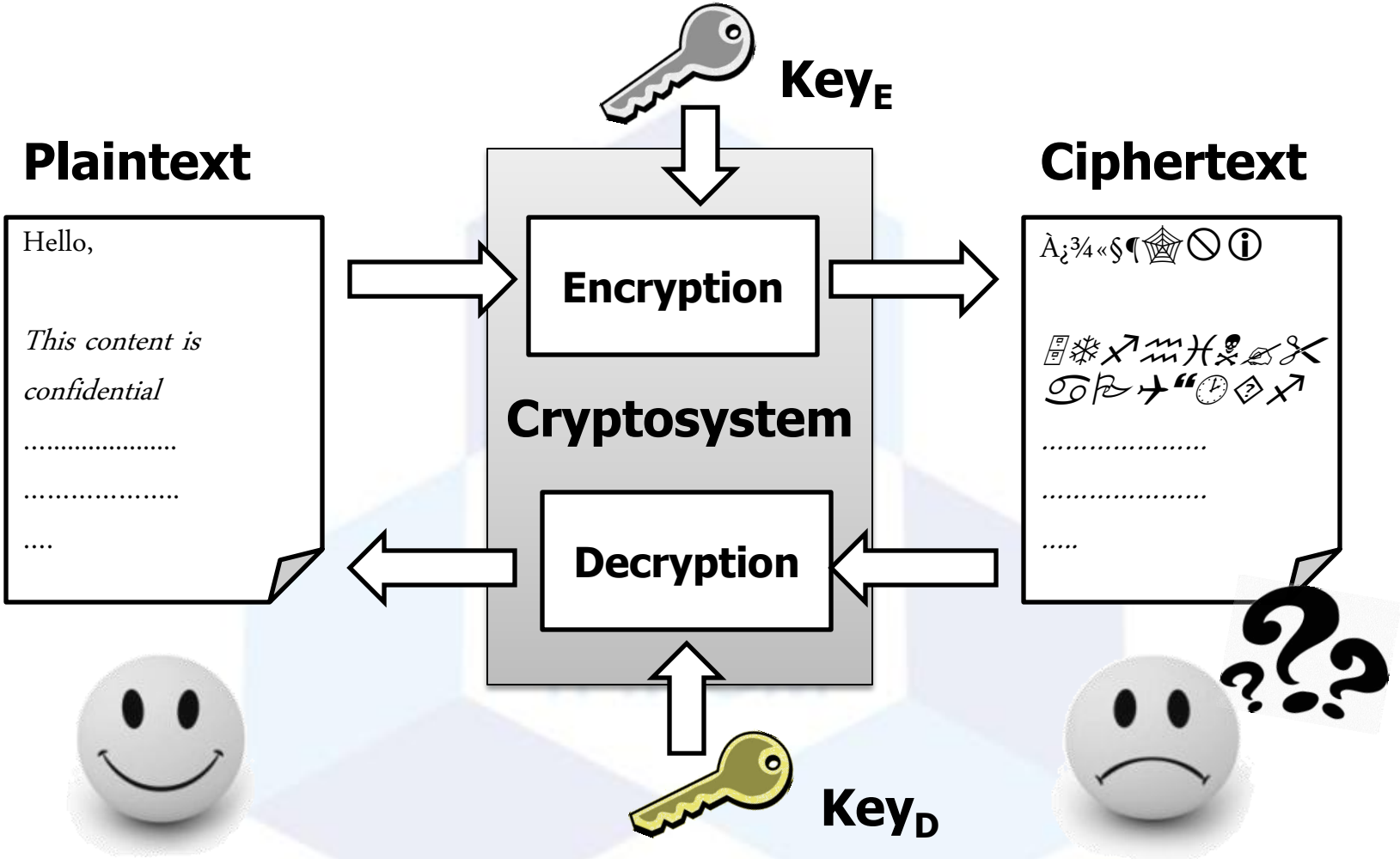
Những khái niệm cơ bản về mã hóa

- Văn bản gốc (plaintext)
- Văn bản mã hóa (ciphertext)
- Hệ thống mã hóa (cryptosystem)
- Khóa (key)
- Hệ thống mã hóa đối xứng (Symmetric cryptosystem)
- Hệ thống mã hóa bất đối xứng (Asymmetric cryptosystem)
- Chữ ký số (Digital signature)
- Chứng thực số (Digital certificate)

Những khái niệm cơ bản về mã hóa

- ***Văn bản gốc (plaintext)*** là văn bản ban đầu có nội dung có thể đọc được và cần được bảo vệ.
- ***Văn bản mã hóa (ciphertext)*** là văn bản sau khi mã hóa, nội dung không thể đọc được.
- ***Mã hóa (encryption)*** là quá trình chuyển văn bản rõ thành văn bản mã hóa. ***Giải mã (decryption)*** là quá trình đưa văn bản mã hóa về lại văn bản gốc ban đầu
- ***Hệ thống mã hóa (cryptosystem)***
  - Cryptosystem = encryption + decryption algorithms
- ***Khóa (key)*** được sử dụng trong quá trình mã hóa và giải mã.

Hệ thống mã hóa

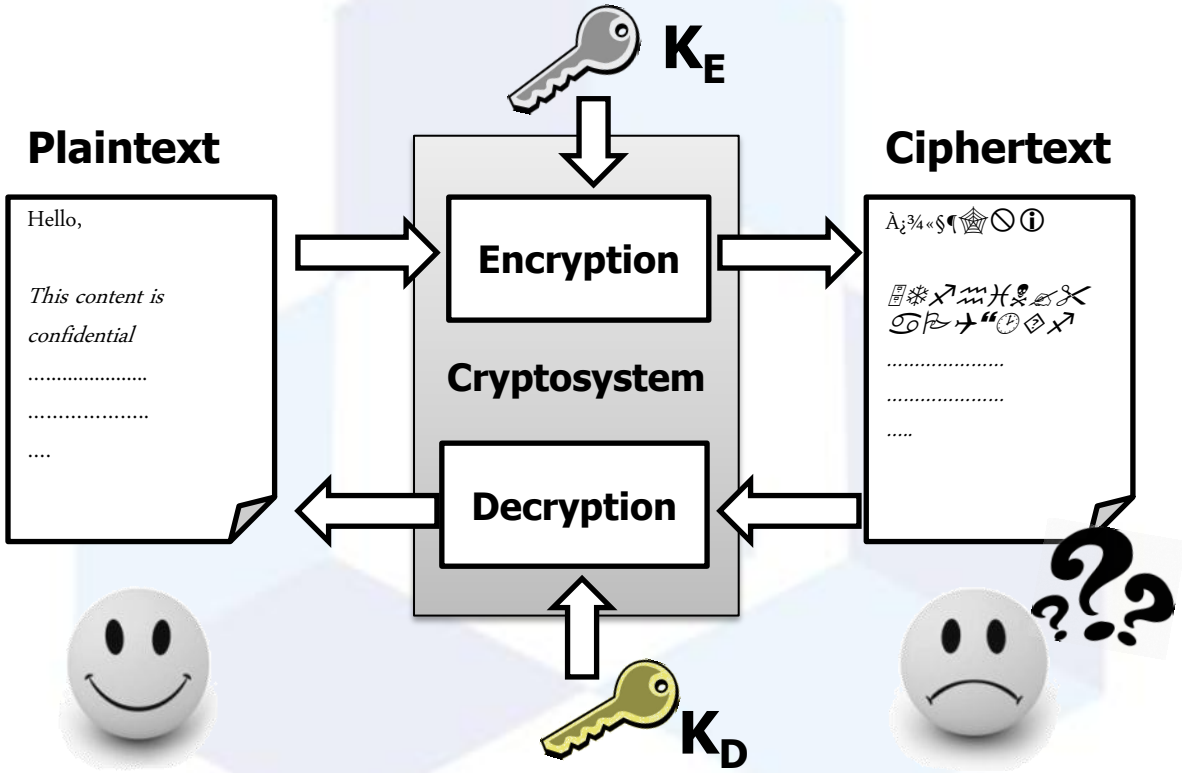


Những khái niệm cơ bản về mã hóa

- ***Hệ thống mã hóa đối xứng (Symmetric cryptosystem)*** là hệ thống mã hóa sử dụng một khóa bí mật chia sẻ (shared-secret-key) cho cả hai quá trình mã hóa và giải mã.
- ***Hệ thống mã hóa bất đối xứng (Asymmetric cryptosystem)*** là hệ thống mã hóa sử dụng một khóa công khai (public key) và một khóa bí mật (private key) cho quá trình mã hóa và giải mã.
  - Hệ thống mã hóa bất đối xứng còn được gọi là hệ thống mã hóa khóa công khai (public-key cryptosystem)

Những khái niệm cơ bản về mã hóa

- Mã hóa đối xứng: $K_E = K_D$
- Mã hóa bất đối xứng: $K_E \neq K_D$



Kỹ thuật mã hóa đối xứng

- Các kỹ thuật mã hóa đối xứng thông dụng: DES, Triple DES, AES
- ***DES: Data Encryption Standard***
  - NBS (National Bureau of Standards) – bây giờ là NIST (National Institute of Standards and Technology) (Mỹ) chọn DES làm tiêu chuẩn mã hóa vào năm 1977.
  - Mỗi thông điệp (message) được chia thành những khối (block) 64 bits
  - Khóa có 56 bits
  - Có thể bị tấn công bằng giải thuật vét cạn khóa (Brute-force or exhaustive key search)

Kỹ thuật mã hóa đối xứng - Triple DES

- 1999, Triple DES được khuyến khích sử dụng thay cho DES
- ***Triple DES***: thực hiện giải thuật DES ba lần.
  - Mã hóa: $c \leftarrow \mathcal{E}_{k1}(D_{k2}(\mathcal{E}_{k1}(m)))$
  - Giải mã: $m \leftarrow D_{k1}(\mathcal{E}_{k2}(D_{k1}(c)))$
    - c: văn bản mã hóa
    - m: văn bản gốc
    - $\mathcal{E}_{k1}(\ )$: mã hóa bằng khóa k1
    - $D_{k1}(\ )$: giải mã bằng khóa k1
  - Triple DES có thể sử dụng các khóa khác nhau.

Kỹ thuật mã hóa đối xứng - AES

- ***AES: Advanced Encryption Standard***
  - Tháng 10/2000, NIST đã chọn AES làm tiêu chuẩn mã hóa thay thế DES
  - AES còn gọi là Rijndael, tên đặt theo hai nhà mật mã học thiết kế ra giải thuật là Daemen và Rijmen
  - Rijndael là giải thuật mã hóa theo khối. Tuy nhiên, khác với DES, Rijndael có thể làm việc với dữ liệu và khóa có độ dài block là 128, 192 hoặc 256 bit.

Kỹ thuật mã hóa bất đối xứng

- Kỹ thuật mã hóa bất đối xứng phổ biến: RSA
- ***RSA***: tên được đặt theo tên 3 nhà phát minh ra giải thuật **R**ivest, **S**hamir và **A**dleman
  - Thuật toán sử dụng 2 khóa có quan hệ toán học với nhau: khóa công khai và khóa bí mật
  - Khóa công khai được công bố rộng rãi cho mọi người và được dùng để mã hóa.
  - Khóa bí mật dùng để giải mã.

So sánh mã hóa đối xứng và bất đối xứng

- Kỹ thuật mã hóa đối xứng có tốc độ mã hóa và giải mã **nhanh hơn** so với kỹ thuật mã hóa bất đối xứng.
- Kỹ thuật mã hóa bất đối xứng **an toàn hơn** so với kỹ thuật mã hóa đối xứng.

So sánh mã hóa đối xứng và bất đối xứng

- Trong thực tế, ta sử dụng kết hợp cả hai kỹ thuật (hybrid scheme) mã hóa đối xứng và bất đối xứng.
  - Kỹ thuật mã hóa bất đối xứng: thích hợp mã hóa những dữ liệu nhỏ và yêu cầu bảo mật cao.

  → Mã hóa khóa bí mật
  - Kỹ thuật mã hóa đối xứng: thích hợp mã hóa những dữ liệu lớn và yêu cầu bảo mật không cao lắm.

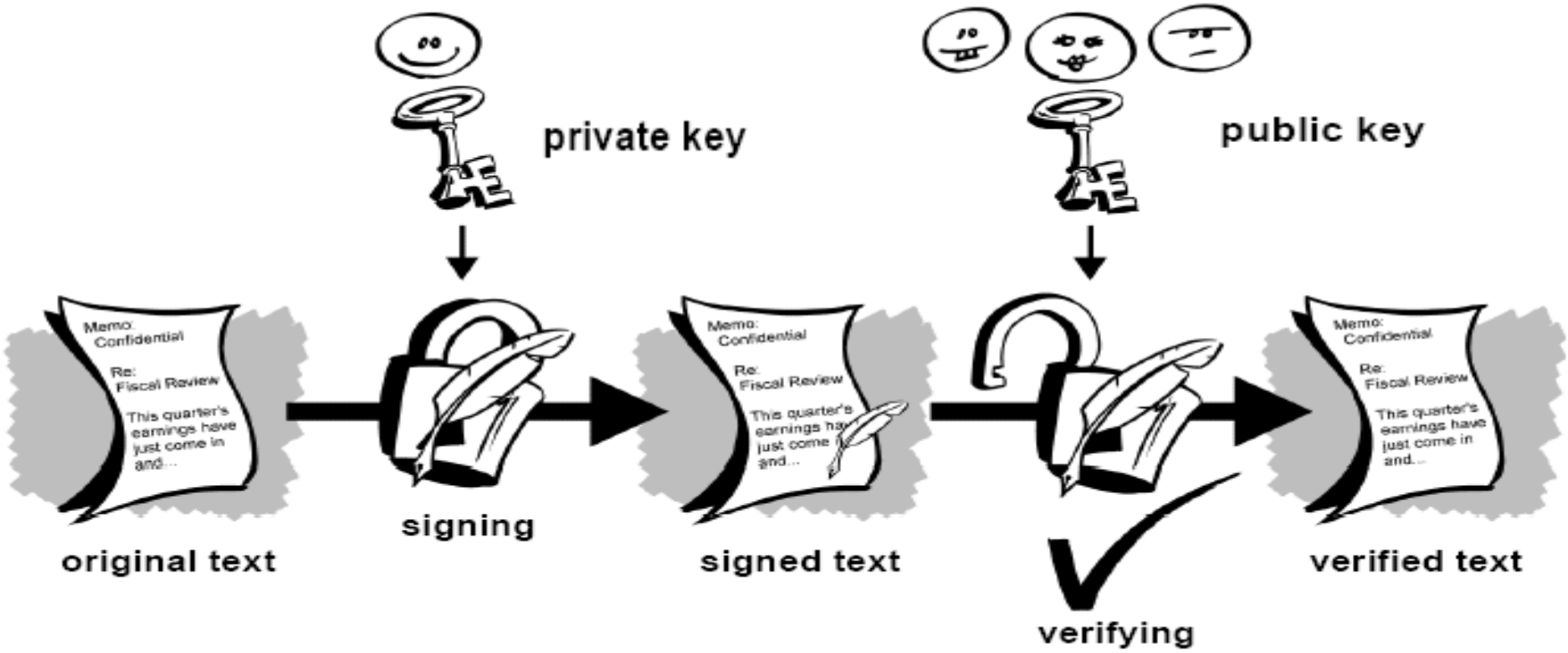
  → Mã hóa dữ liệu

Chữ ký số

- ***Chữ ký số (Digital signature)***: là thông điệp (có thể là văn bản, hình ảnh, hoặc video...) đã được ***ký*** bằng khóa bí mật của người dùng nhằm mục đích xác định người chủ của thông điệp đó.
- Mục đích của chữ ký số:
  - Xác thực: xác định ai là chủ của thông điệp
  - Tính toàn vẹn : kiểm tra xem thông điệp có bị thay đổi
  - Tính chống thoái thác: ngăn chặn việc người dùng từ chối đã tạo ra và gửi thông điệp

Chữ ký số

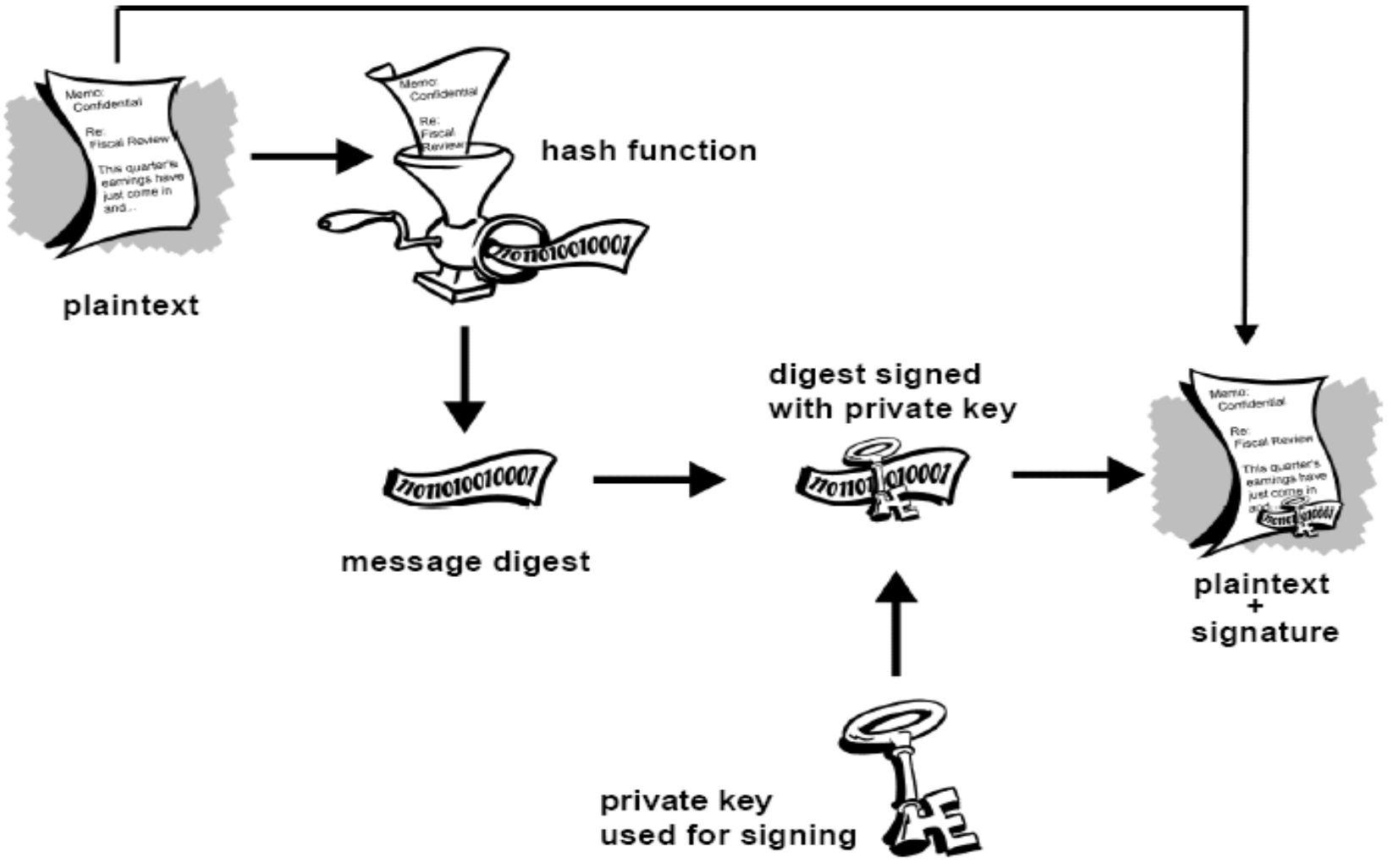
- Chữ ký số:
  - Dùng khóa bí mật để ký (mã hóa) lên thông điệp → chữ ký
  - Dùng khóa công khai để xác thực (giải mã) chữ ký



Chữ ký số

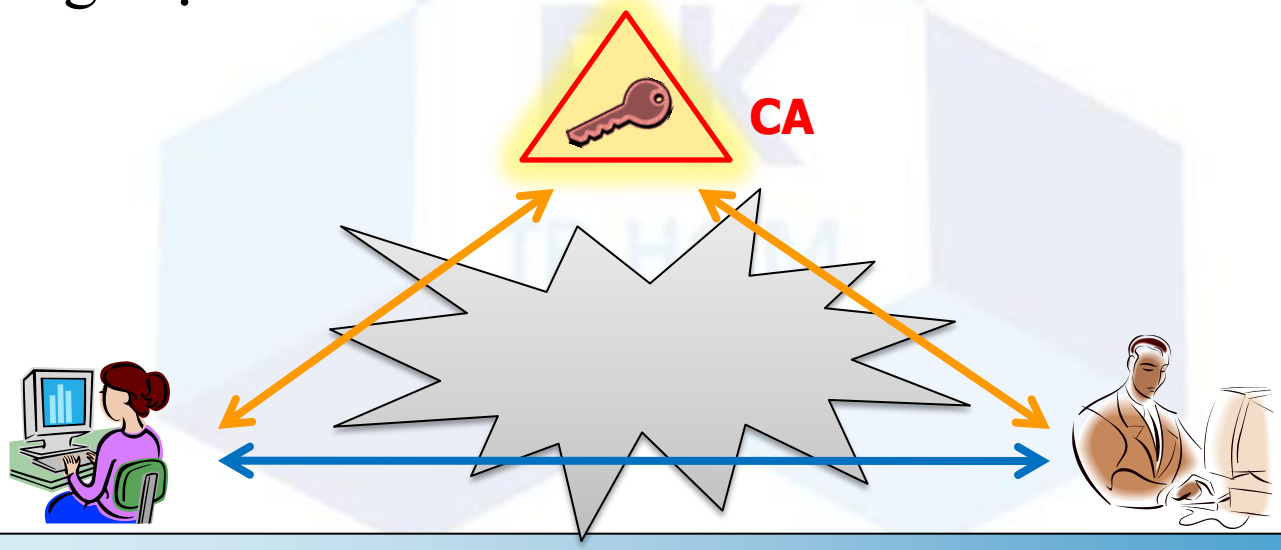
- Quá trình đơn giản của chữ ký số
  - Alice viết một văn bản và muốn gửi cho Bob
  - Alice **ký** lên văn bản bằng **khóa bí mật** → Văn bản đã ký
  - Alice gửi **văn bản gốc và văn bản đã ký** cho Bob qua đường truyền mạng
  - Bob nhận được văn bản gốc và văn bản đã ký
  - Bob dùng **khóa công khai** của Alice để **giải mã** văn bản đã ký
  - Bob **so sánh** văn bản giải mã được và văn bản gốc, nếu giống nhau thì đây chính là do Alice gửi, nếu sai thì đây không phải văn bản do Alice gửi.

Chữ ký số an toàn (Secure digital signature)



Chứng thực số

- ***Chứng thực số (digital certificate)***, hoặc chứng thực khóa công khai (public key certificate), là một tài liệu điện tử dùng để xác minh một khóa công khai là của ai.
- Trong mô hình hạ tầng khóa công khai (public key infrastructure), CA (Certificate Authority) là nhà cung cấp chứng thực số.



Chứng thực số

- Mỗi chứng thực số bao gồm các thông tin cơ bản sau:
  - Tên và URL của CA cung cấp chứng thực
  - Khóa công khai
  - Tên sở hữu: cá nhân, tổ chức, máy chủ
  - Thời hạn sử dụng
- CA sẽ chịu trách nhiệm ký lên mỗi chứng thực số

Nội dung

1. Những khái niệm cơ bản về mã hóa
2. Mã hóa hoàn hảo
3. Kênh trao đổi khóa
4. Mô hình Dolev-Yao
5. Giao thức trao đổi khóa

Mã hóa hoàn hảo (Perfect encryption)

- Ký hiệu:
  - M: văn bản gốc
  - A: giải thuật mã hóa
  - K: khóa mã hóa
  - M': văn bản mã hóa
  - A': giải thuật giải mã
  - K': khóa giải mã
- Mối quan hệ của M và M' được biểu diễn như sau:

$$M' = A(K,M) = \{M\}_K$$

$$M = A'(K',M') = A'(K', A(K,M))$$

Mã hóa hoàn hảo

- $M' = A(K,M) = \{M\}_K$
- Một giải thuật mã hóa được gọi là hoàn hảo cần phải đảm bảo các tính chất sau:
- *Tính chất 1:* Nếu không có khóa bí mật K (trong mã hóa đối xứng), hoặc khóa bí mật K' tương ứng với K (trong mã hóa bất đối xứng), thì không có cách nào có thể tìm ra được văn bản gốc M từ văn bản mã hóa $\{M\}_K$

Mã hóa hoàn hảo

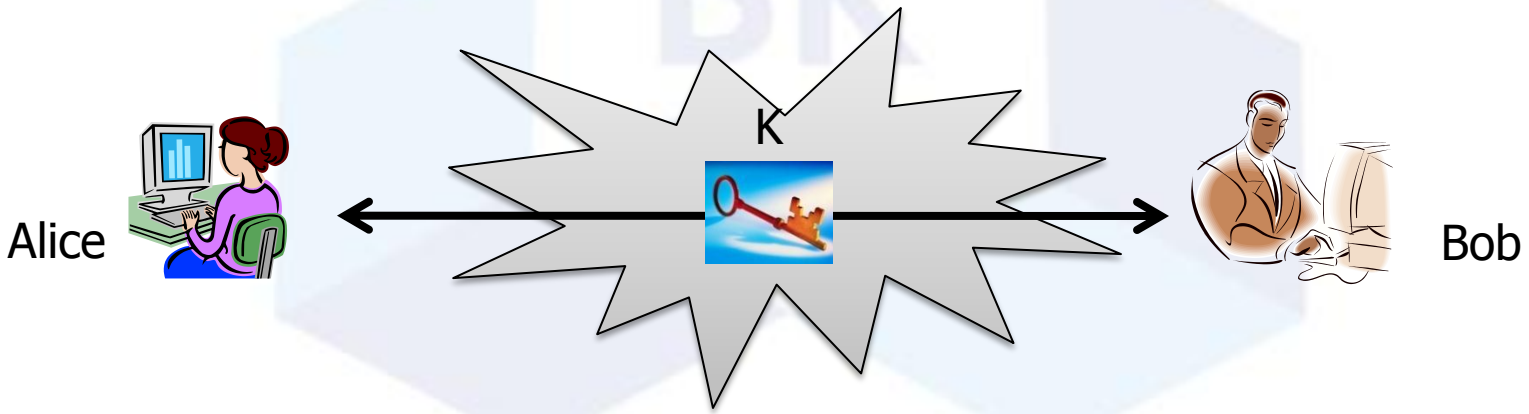
- $M' = A(K,M) = \{M\}_K$
- Một giải thuật mã hóa được gọi là hoàn hảo cần phải đảm bảo các tính chất sau (tiếp theo):
- *Tính chất 2:* Nếu có văn bản mã hóa $\{M\}_K$ và một phần thông tin về văn bản gốc M, thì cũng không có cách nào có thể tìm ra được khóa bí mật K (trong mã hóa đối xứng), hoặc khóa bí mật K' tương ứng với K (trong mã hóa bất đối xứng)
- *Tính chất 3:* Nếu không có khóa K thì dù có thông tin của văn bản gốc M cũng không thể thay đổi $\{M\}_K$ mà không bị phát hiện trong quá trình giải mã.

Nội dung

1. Những khái niệm cơ bản về mã hóa
2. Mã hóa hoàn hảo
3. Kênh trao đổi khóa
4. Mô hình Dolev-Yao
5. Giao thức trao đổi khóa

Kênh trao đổi khóa

- Giả sử Alice và Bob muốn nói chuyện một cách bí mật với nhau thông qua kỹ thuật mã hóa đối xứng.
- Alice và Bob chưa từng thỏa thuận với nhau về một khóa bí mật chung.
- Kênh trao đổi khóa là nơi/cách thức/kỹ thuật mà Alice dùng để trao đổi đổi với nhau về khóa bí mật chung.



Kênh trao đổi khóa

- Cách truyền thống: sử dụng dịch vụ xác thực trực tuyến (online authentication service). Đây là dịch vụ xác thực thông qua một bên thứ ba tin cậy (Trusted Third Party – TTP)
- Sử dụng kỹ thuật mã hóa công khai
- Sử dụng kỹ thuật phân phối khóa lượng tử (Quantum Key Distribution Technique) *(đọc thêm mục [4]4.4.5)*

Kênh trao đổi khóa

- Những tính chất bảo mật cần có của một kênh trao đổi khóa:
  1. Chỉ Alice và Bob (và có thể một bên đáng tin cậy khác, TTP) biết khóa bí mật K
  2. Alice và Bob biết chắc rằng người kia cũng biết khóa K
  3. Alice và Bob biết chắc rằng khóa K là khóa mới được tạo ra

Nội dung

1. Những khái niệm cơ bản về mã hóa
2. Mã hóa hoàn hảo
3. Kênh trao đổi khóa
4. Mô hình Dolev-Yao
5. Giao thức trao đổi khóa

Mô hình Dolev-Yao

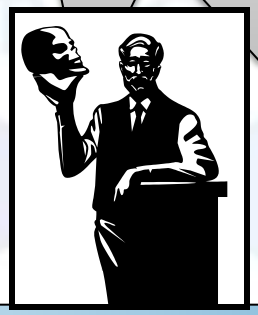
- Mô hình Dolev-Yao mô tả các mối nguy hiểm được dùng để đánh giá các giao thức mã hóa.
- Mô hình Dolev-Yao có 4 nhân vật:
  - **Alice** và **Bob**: là 2 người dùng bình thường và muốn thực hiện một cuộc nói chuyện bí mật và an toàn.
  - **Trent**: là một người trung gian đáng tin cậy (Trusted Third Party)
  - **Malice**: là người xấu có ý muốn phá, nghe trộm, hoặc giả mạo nội dung cuộc nói chuyện giữa Alice và Bob

Mô hình Dolev-Yao

Trent

Alice

Bob



Malice

Mô hình Dolev-Yao

- Mô hình Dolev-Yao định nghĩa các việc Malice có thể và không thể làm.
- Malice (có thể):
  - Malice là một người dùng hợp lệ của hệ thống, do vậy Malice có thể bắt đầu một cuộc nói chuyện bình thường với các người dùng khác.
  - Xem bất kỳ thông điệp nào được truyền qua môi trường mạng
  - Sẽ có cơ hội trở thành người nhận thông điệp từ bất kỳ người dùng nào.

Có thể mạo danh một người dùng bất kỳ gửi thông điệp đến một người dùng bất kỳ khác.

Mô hình Dolev-Yao

- Malice (không thể):
  - Không thể đoán một số ngẫu nhiên từ một không gian đủ lớn.
  - Không thể giải mã ra được văn bản gốc từ văn bản mã hóa nếu không có khóa bí mật K
  - Không thể tạo ra được một văn bản mã hóa hợp lệ từ một văn bản gốc cho trước nếu không có khóa đúng
  - Không thể suy ra được khóa bí mật từ khóa công khai.
  - Malice có thể điều khiển và truy cập những thành phần/thiết bị chung của hệ thống; nhưng không thể điều khiển và truy cập những thành phần/thiết bị cá nhân của các người dùng khác, như bộ nhớ của máy tính cá nhân.

Mô hình Dolev-Yao

- Giả sử Alice và Bob muốn trao đổi với nhau một cách bí mật và an toàn.
- Giả sử Alice và Bob chưa từng gặp nhau trước đó, do vậy họ chưa có thỏa thuận một khóa bí mật dùng chung, và cũng chưa có thông tin gì về khóa công khai của người kia.
- Vậy làm sao Alice và Bob có thể trao đổi với nhau một cách an toàn và bí mật qua một môi trường mạng không an toàn?

Nội dung

1. Những khái niệm cơ bản về mã hóa
2. Mã hóa hoàn hảo
3. Kênh trao đổi khóa
4. Mô hình Dolev-Yao
5. Giao thức trao đổi khóa

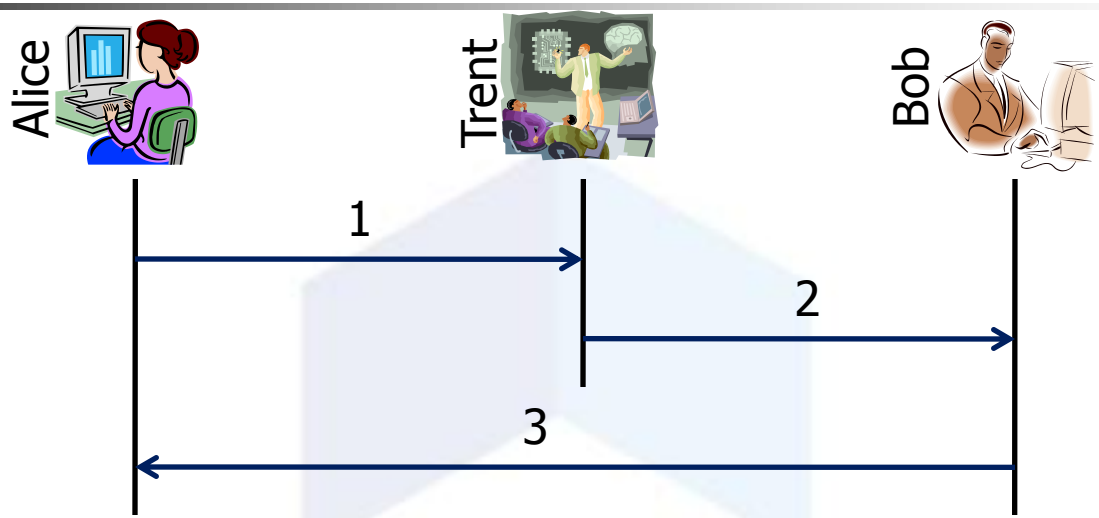
Giao thức trao đổi khóa

- Giao thức "From Alice to Bob"
- Giao thức "Session Key from Trent"
- Giao thức "Message Authentication"
- Giao thức "Challenge-response"
- Giao thức dùng mã hóa công khai

Giao thức "From Alice to Bob"

- Giả sử:
  - Alice và Trent đã có một khóa bí mật chung K_{AT}
  - Bob và Trent cũng có một khóa bí mật chung K_{BT}
- Mục tiêu: Alice và Bob muốn thiết lập một khóa phiên (session key) bí mật chung mới K để nói chuyện
- Trong giao thức này, Alice là người tạo khóa phiên, thông qua Trent làm trung gian, và gửi cho Bob

Giao thức "From Alice to Bob"



1. Alice tạo ra khóa K ngẫu nhiên; mã hóa $\{K\}_{KAT}$; và gửi cho Trent:
   Alice, Bob, $\{K\}_{KAT}$
2. Trent tìm khóa K_{AT}, K_{BT}; giải mã $\{K\}_{KAT}$ để lấy K rồi mã hóa lại $\{K\}_{KBT}$; và gửi cho Bob: *Alice, Bob,* $\{K\}_{KBT}$
3. Bob giải mã $\{K\}_{KBT}$ để lấy K; và bắt đầu nói chuyện với Alice:
   $\{$*Hello Alice, I'm Bob!*$\}_K$

Giao thức "From Alice to Bob"

- Vấn đề của giao thức "From Alice to Bob" là khóa K được tạo bởi Alice có thể không đủ an toàn (chiều dài khóa không đủ dài hoặc dễ bị đoán ra)
- Bob có thể cảm thấy không an toàn khi sử dụng khóa K và từ chối nói chuyện.

→ Giao thức mới: "Session key from Trent"

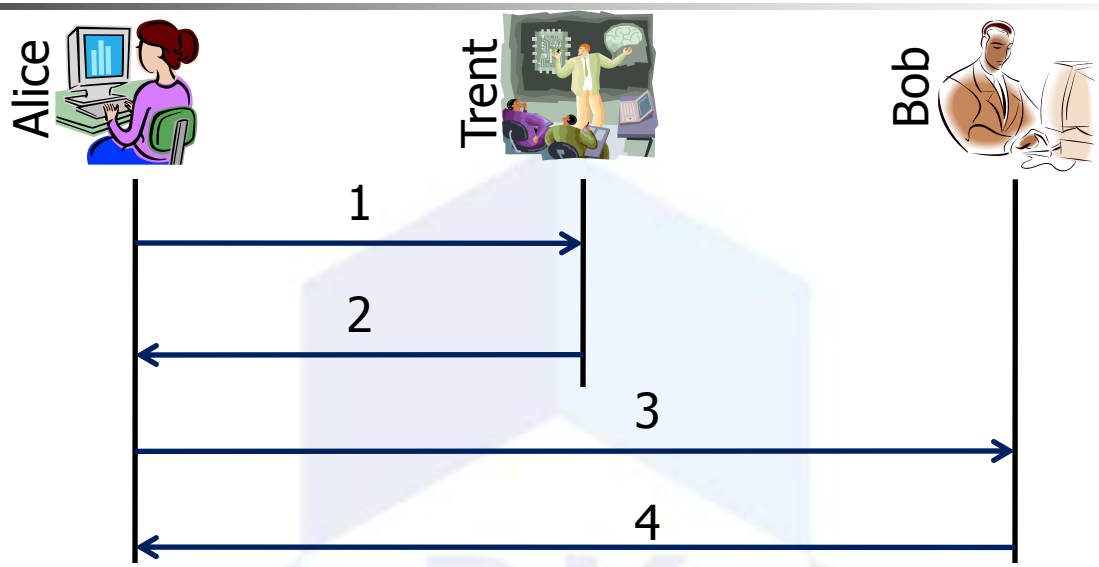
Giao thức trao đổi khóa

- Giao thức "From Alice to Bob"
- **Giao thức "Session Key from Trent"**
- Giao thức "Message Authentication"
- Giao thức "Challenge-response"
- Giao thức dùng mã hóa công khai

Giao thức "Session Key from Trent"

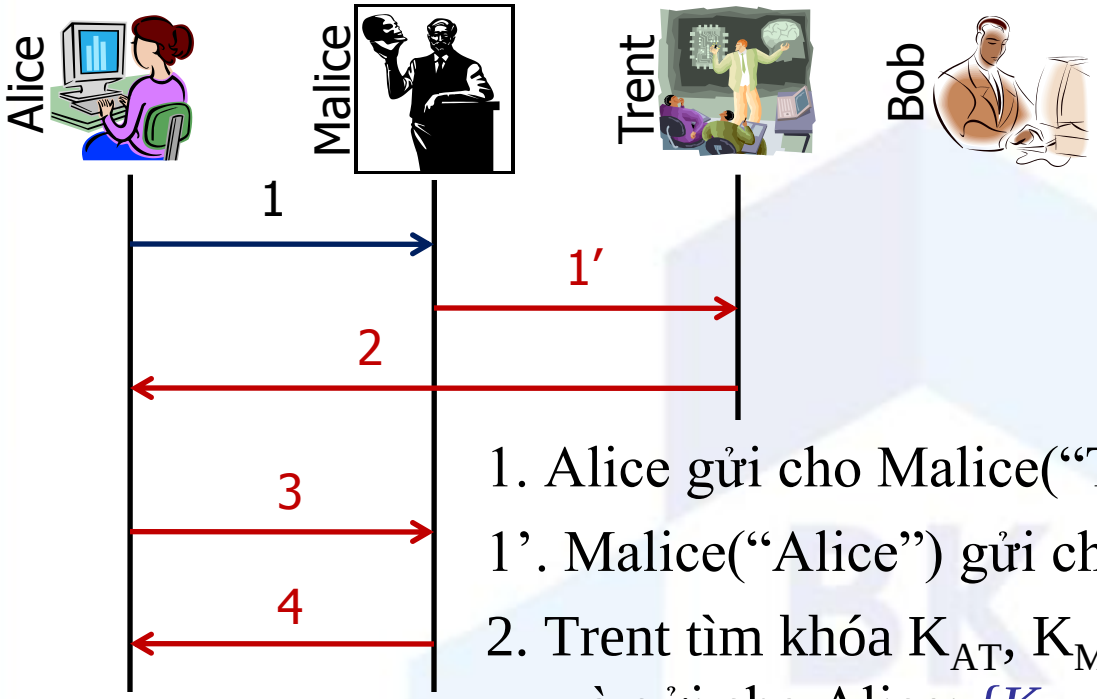
- Giả sử:
  - Alice và Trent đã có một khóa bí mật chung K_{AT}
  - Bob và Trent cũng có một khóa bí mật chung K_{BT}
- Mục tiêu: Alice và Bob muốn thiết lập một khóa bí mật chung mới K để nói chuyện
- Trong giao thức này, khóa phiên do Trent tạo ra

Giao thức "Session Key from Trent"



1. Alice gửi cho Trent: *Alice, Bob*
2. Trent tìm khóa K_{AT}, K_{BT}; tạo khóa K ngẫu nhiên; và gửi cho Alice: $\{K\}_{KAT}, \{K\}_{KBT}$
3. Alice giải mã $\{K\}_{KAT}$; và gửi cho Bob: *Trent, Alice,* $\{K\}_{KBT}$
4. Bob giải mã $\{K\}_{KBT}$ được K; và bắt đầu nói chuyện với Alice: $\{Hello\ Alice,\ I'm\ Bob!\}_K$

Tấn công giao thức "Session key from Trent"



K_{MT}: khóa bí mật chung giữa Malice và Trent

1. Alice gửi cho Malice("Trent"): *Alice, Bob*
1'. Malice("Alice") gửi cho Trent: *Alice, Malice*
2. Trent tìm khóa K_{AT}, K_{MT}; tạo ra khóa K_{AM} ngẫu nhiên; và gửi cho Alice: $\{K_{AM}\}_{KAT}$, $\{K_{AM}\}_{KMT}$
3. Alice giải mã $\{K_{AM}\}_{KAT}$ và gửi cho Malice("Bob"): *Trent, Alice,* $\{K_{AM}\}_{KMT}$
4. Malice("Bob") gửi cho Alice: $\{$*Hello Alice, I'm Bob!*$\}_{KAM}$

Tấn công giao thức "Session key from Trent"

- Kết quả của tấn công trên:
  - Alice tưởng rằng đang trao đổi khóa chung với Bob, nhưng thật ra là với Malice
  - Malice giả mạo Bob nói chuyện với Alice
  - Bob không tham gia vào cuộc nói chuyện
- Vấn đề của giao thức "Session key from Trent"
  - Malice là một người dùng hợp lệ trong hệ thống, và Trent cũng xem Malice như một người dùng bình thường
  - Những người tấn công từ bên trong thường nguy hiểm hơn những người bên ngoài

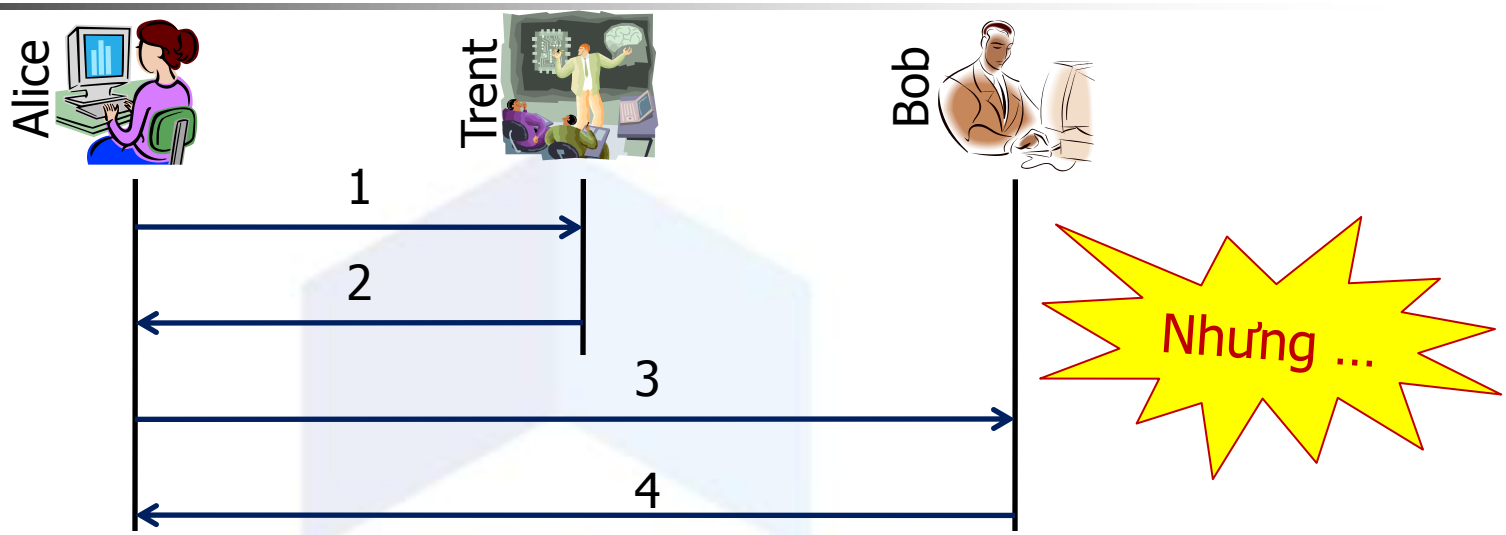
Khắc phục giao thức "Session key from Trent"

- Để khắc phục trường hợp tấn công trên, giao thức được chỉnh sửa lại ở bước 1 như sau:

  1. Alice sends to Trent: $Alice, \{Bob\}_{K_{AT}}$

  - Tên của Bob được mã hóa bằng K_{AT}
  - Tại sao tên của Alice không mã hóa?

Khắc phục giao thức "Session Key from Trent"



1. Alice gửi cho Trent: *Alice*, $\{Bob\}_{KAT}$
2. Trent tìm khóa K_{AT} và giải mã $\{Bob\}_{KAT}$ để biết người mà Alice muốn tạo khóa chung; Trent tìm khóa K_{BT}; tạo khóa K ngẫu nhiên; và gửi cho Alice: $\{K\}_{KAT}$, $\{K\}_{KBT}$
3. Alice giải mã $\{K\}_{KAT}$; và gửi cho Bob: *Trent, Alice,* $\{K\}_{KBT}$
4. Bob giải mã $\{K\}_{KBT}$ được K; và bắt đầu nói chuyện với Alice: $\{Hello\ Alice,\ I'm\ Bob!\}_K$

Tấn công giao thức "Session key from Trent"

- Malice có thể tấn công như sau:
  1. Alice sends to Trent: *Alice*, $\{Bob\}_{KAT}$
  1'. Malice("Alice") sends to Trent: *Alice*, $\{Malice\}_{KAT}$
- Tại sao?
  - Malice có thể tạo được $\{Malice\}_{KAT}$
  - Malice biết được Bob là người Alice muốn nói chuyện
- Kết quả: Malice đóng giả Bob nói chuyện với Alice

Tấn công giao thức "Session key from Trent"

- Một cách tấn công khác:
  - Trong lần nói chuyện hợp lệ trước đó với Alice, Malice đã lưu lại khóa K' và $\{K'\}_{KAT}$
  - Malice có thể sử dụng lại một khóa cũ K' và $\{K'\}_{KAT}$

1. Alice gửi cho Malice("Trent"): $Alice, \{Bob\}_{KAT}$

2'. Malice("Trent") gửi cho Alice: $\{\mathbf{K'}\}_{K_{AT}}$, …

- Kết quả: Malice đóng giả Bob nói chuyện với Alice bằng khóa K' cũ

Giao thức "Session key from Trent"

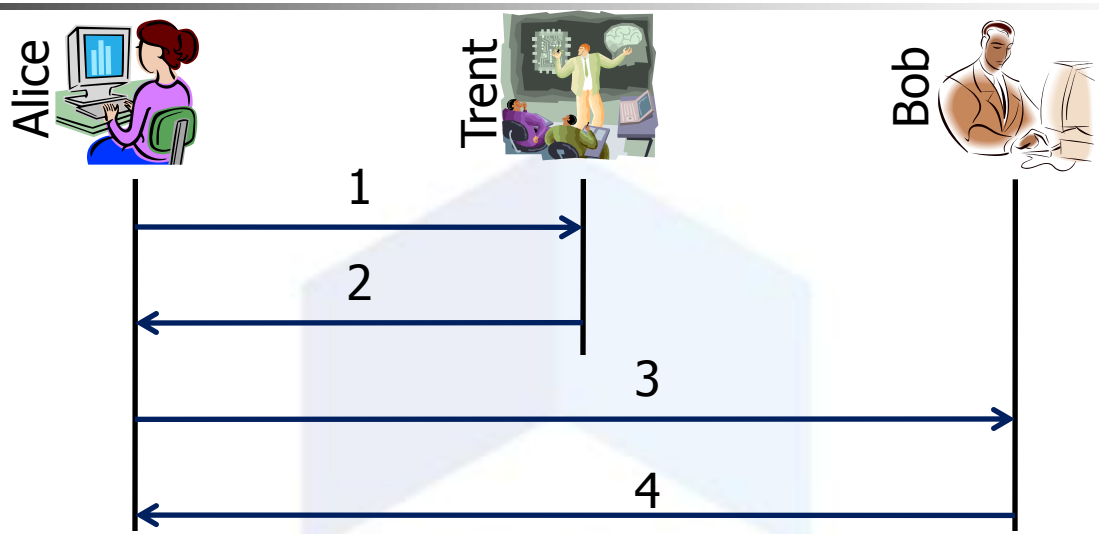
- Malice có thể chỉnh sửa các thông điệp trong giao thức mà không bị phát hiện.
- Do vậy giao thức cần một dịch vụ bảo mật có thể chống lại việc thay đổi các thông điệp trong giao thức.

→ Giao thức "Message Authentication"

Giao thức trao đổi khóa

- Giao thức "From Alice to Bob"
- Giao thức "Session Key from Trent"
- **Giao thức "Message Authentication"**
- Giao thức "Challenge-response"
- Giao thức dùng mã hóa công khai

Giao thức "Message Authentication"



1. Alice gửi cho Trent: *Alice, Bob*
2. Trent tìm khóa K_{AT}, K_{BT}; tạo khóa K ngẫu nhiên; và gửi cho Alice: $\{Bob, K\}_{KAT}$, $\{Alice, K\}_{KBT}$
3. Alice giải mã $\{Bob, K\}_{KAT}$ và kiểm tra danh định của Bob; rồi gửi cho Bob: *Trent,* $\{Alice, K\}_{KBT}$
4. Bob giải mã $\{Alice, K\}_{KBT}$ và kiểm tra danh định của Alice; bắt đầu nói chuyện với Alice: $\{Hello\ Alice,\ I'm\ Bob!\}_K$

Mã hóa hoàn hảo và giao thức "Message Authentication"

- *Dựa vào tính chất 3 của mã hóa hoàn hảo:* Nếu không có khóa K thì dù có thông tin của văn bản gốc M cũng không thể thay đổi $\{M\}_K$ mà không bị phát hiện trong quá trình giải mã.

  → Malice không thể chỉnh sửa khối văn bản mã hóa $\{Bob, K\}_{KAT}$ và $\{Alice, K\}_{KBT}$ mà không bị phát hiện.

Tấn công giao thức "Message Authentication"

- Tấn công bằng cách lặp lại thông điệp (message replay attack)
- Malice chặn lại thông điệp của Alice và sửa thành:

1. Alice gửi cho Malice("Trent"): *Alice, Bob*
2. Malice("Trent") gửi cho Alice: $\{Bob,K'\}K_{AT},\{Alice,K'\}K_{BT}$

- Hai khối văn bản mã hóa chứa K' được Malice lưu lại trong lần thực hiện giao thức của cuộc nói chuyện trước đó giữa Alice và Bob.
- Cách tấn công này sẽ làm cho Alice và Bob sử dụng lại khóa phiên K' cũ.
- Vì K' là khóa cũ nên Malice có thể tìm ra được giá trị K' *(bằng cách nào?)*

Giao thức trao đổi khóa

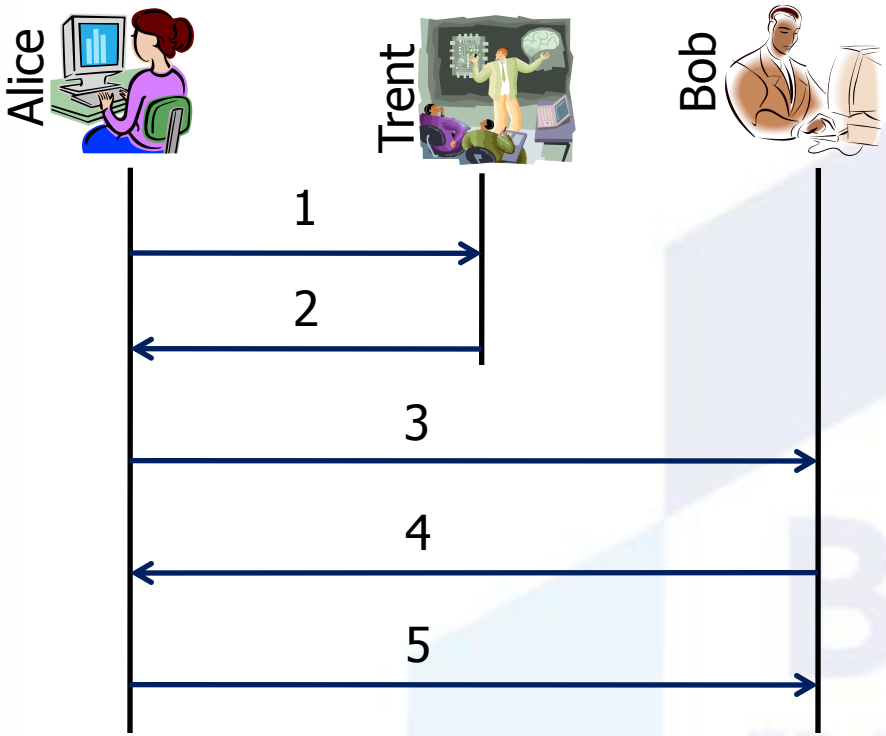
- Giao thức "From Alice to Bob"
- Giao thức "Session Key from Trent"
- Giao thức "Message Authentication"
- **Giao thức "Challenge-response"**
- Giao thức dùng mã hóa công khai

Giao thức "Challenge-response"

- Giao thức "Challenge-response" bổ sung thêm một số bước nhằm giúp cho Alice và Bob xác nhận một khóa phiên có mới hay không.

  → Chống lại message replay attack
- Giao thức này được Needham và Schroeder đề nghị năm 1978 và còn được gọi là giao thức "Needham and Schroeder"
- Giao thức sử dụng số Nonce (a number used once) – số chỉ được sử dụng 1 lần

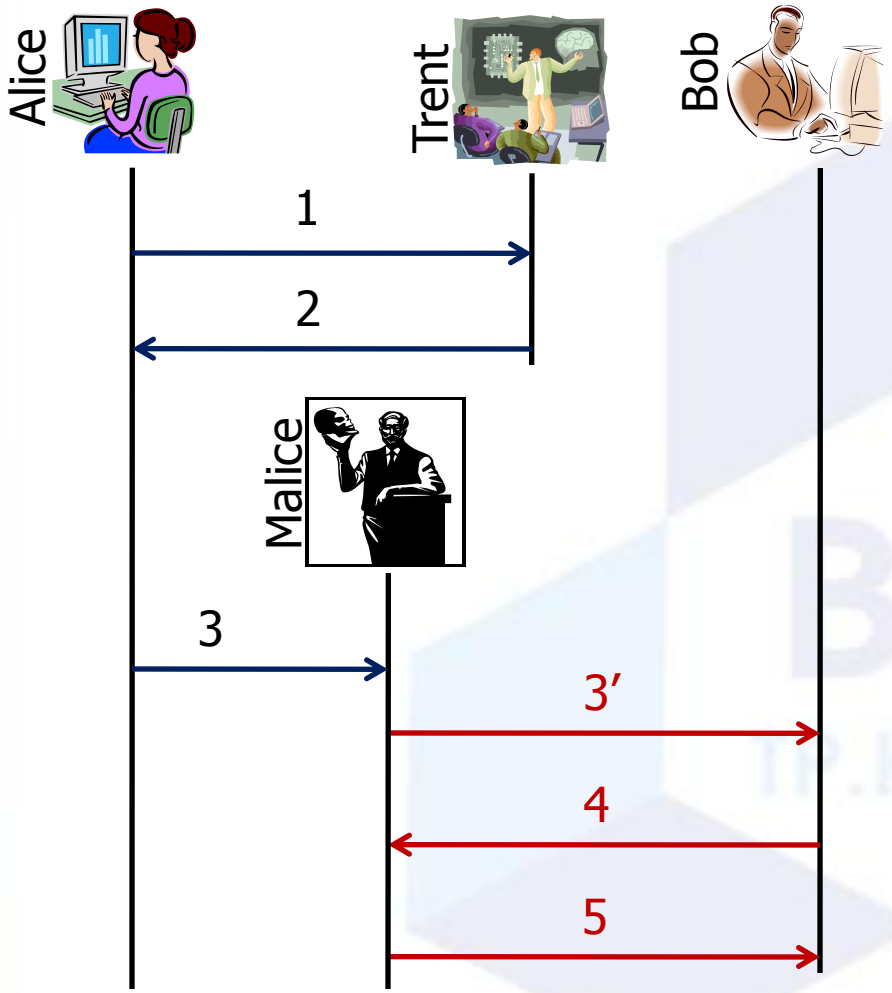
Giao thức "Challenge-response"



N_A/N_B: số Nonce tạo bởi Alice/Bob

1. Alice tạo số N_A ngẫu nhiên và gửi cho Trent: *Alice, Bob, N_A*
2. Trent tạo khóa K ngẫu nhiên và gửi cho Alice: *$\{N_A, K, Bob, \{K, Alice\}_{KBT}\}_{KAT}$*
3. Alice giải mã, kiểm tra số N_A, kiểm tra danh định của Bob, và gửi cho Bob: *Trent, $\{K, Alice\}_{KBT}$*
4. Bob giải mã, kiểm tra danh định của Alice, tạo số N_B ngẫu nhiên và gửi cho Alice: *$\{I'm\ Bob!\ N_B\}_K$*
5. Alice gửi cho Bob: *$\{I'm\ Alice!N_B\text{-}1\}_K$*

Tấn công giao thức "Challenge-response"



1. Alice gửi cho Trent: *Alice, Bob,* N_A
2. Trent gửi cho Alice: $\{N_A, K, Bob, \{K, Alice\}_{KBT}\}_{KAT}$
3. Alice gửi cho Malice("Bob"): *Trent,* $\{K, Alice\}_{KBT}$

3'. Malice("Alice") gửi cho Bob: *Trent,* $\{K', Alice\}_{KBT}$

4. Bob giải mã, kiểm tra danh định của Alice, tạo số N_B ngẫu nhiên và gửi cho Malice("Alice"): $\{I'm\ Bob!\ N_B\}_{K'}$
5. Malice("Alice") gửi cho Bob: $\{I'm\ Alice!N_B\text{-}1\}_{K'}$

Tấn công giao thức "Challenge-response"

- Kết quả của tấn công này là:
  - Bob nghĩ rằng mình đang trao đổi một khóa phiên mới với Alice nhưng thật ra đây là một khóa phiên cũ và Malice có thể biết khóa cũ này.
  - Malice đóng giả Alice để nói chuyện với Bob
  - Alice không thực hiện thành công cuộc nói chuyện với Bob

Giao thức "Challenge-response"

- Alice dựa vào số N_A để xác định thông điệp đúng là do Trent gửi và khóa phiên là mới (bước 1-2-3)
- Bob không có cơ sở để xác định khóa phiên là do Trent tạo ra và là khóa phiên mới

  → Bổ sung thêm một số thông điệp trao đổi giữa Bob và Trent

  → Sử dụng Timestamps

Giao thức "Challenge-response" với Timestamps

1. Alice gửi cho Trent: *Alice, Bob*
2. Trent gửi cho Alice: $\{Bob, K, T, \{Alice, K, T\}_{KBT}\}_{KAT}$
3. Alice kiểm tra T và gửi cho Bob: $\{Alice, K, T\}_{KBT}$
4. Bob kiểm tra T và gửi cho Alice: $\{I'm\ Bob!\ N_B\}_K$
5. Alice gửi cho Bob: $\{I'm\ Alice! N_B\text{-}1\}_K$

- Kiểm tra T: $|Clock - T| < \Delta t_1 + \Delta t_2$
  - Clock: đồng hồ tại máy cá nhân
  - T: timestamp, giờ tại Trent
  - Δt_1 , Δt_2 : độ lệch múi giờ và độ lệch thời gian cho phép
- Không được áp dụng do khó có thể điều chỉnh giờ chuẩn rộng rãi.

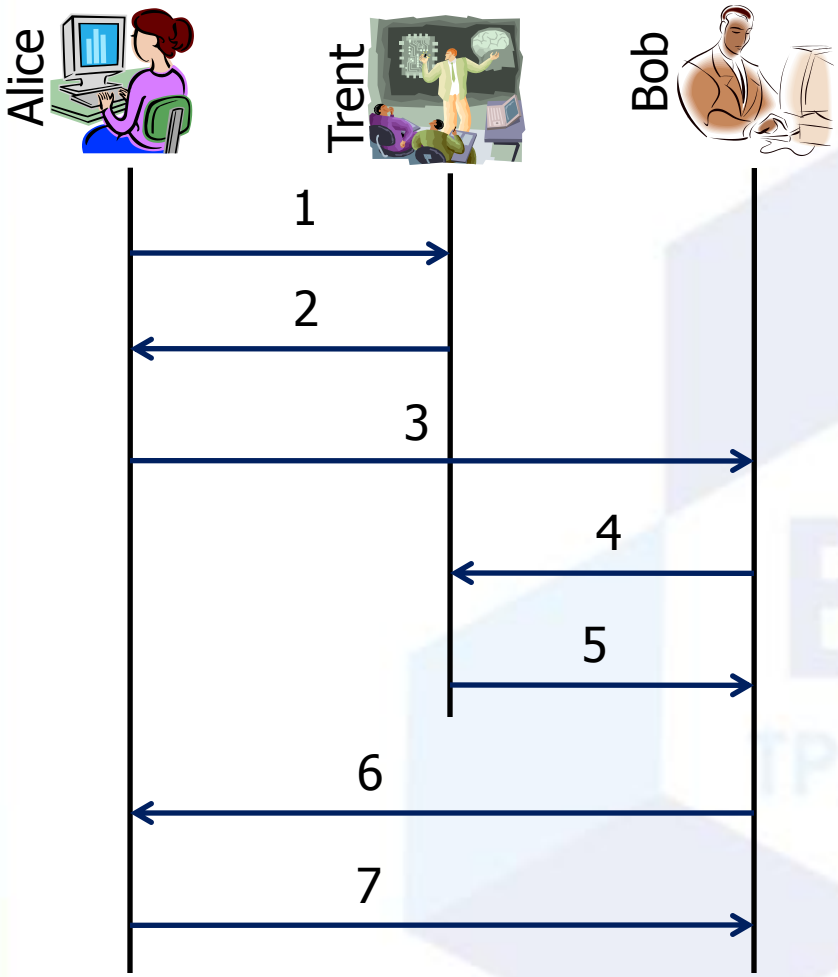
Giao thức trao đổi khóa

- Giao thức "From Alice to Bob"
- Giao thức "Session Key from Trent"
- Giao thức "Message Authentication"
- Giao thức "Challenge-response"
- **Giao thức dùng mã hóa công khai**

Giao thức dùng Mã hóa công khai

- K_A, K^{-1}_A: là khóa công khai và khóa bí mật của Alice
- K_B, K^{-1}_B: là khóa công khai và khóa bí mật của Bob
- K_M, K^{-1}_M: là khóa công khai và khóa bí mật của Malice
- K_T, K^{-1}_T: là khóa công khai và khóa bí mật của Trent
- $\{M\}_{KA}$: mã hóa M bằng khóa công khai K_A
- $\{M\}_{K^{-1}_A}$: ký lên M bằng khóa bí mật K_A

Giao thức dùng Mã hóa công khai



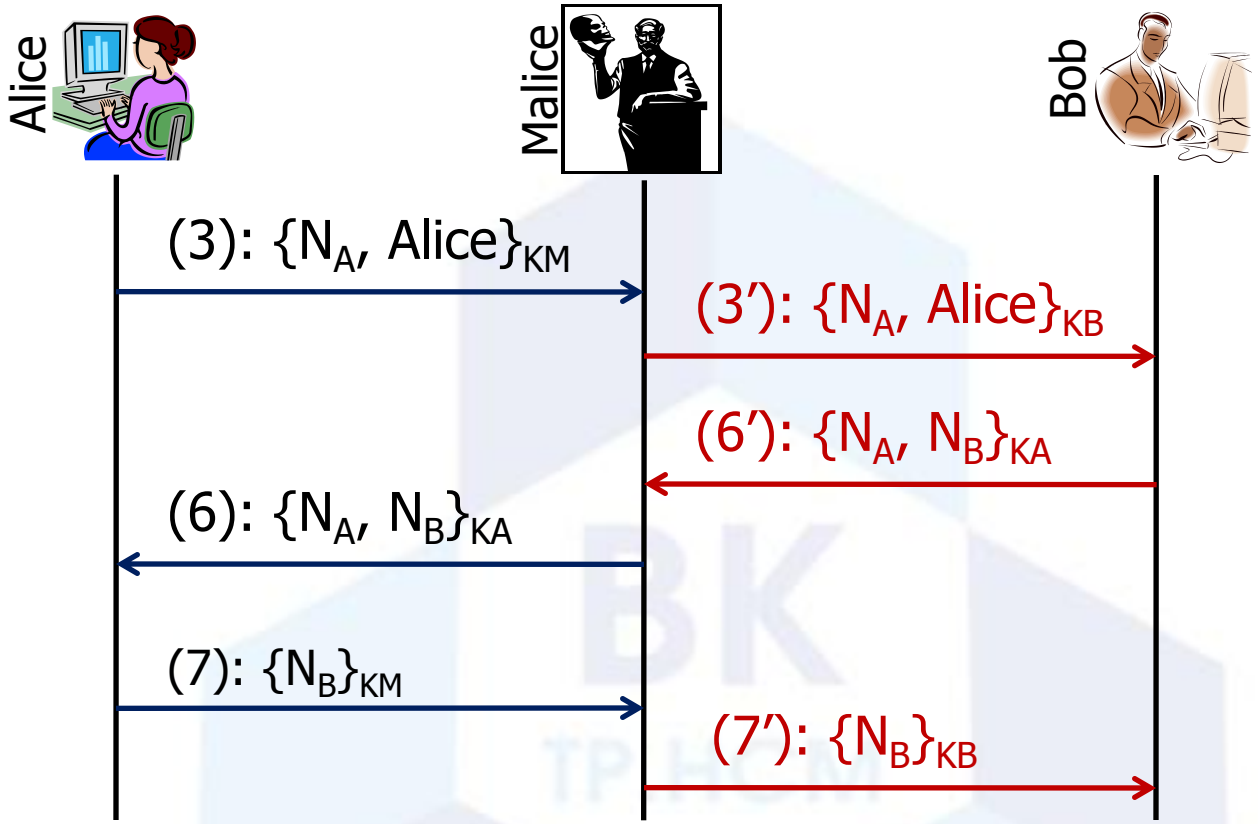
1. Alice gửi cho Trent: *Alice, Bob*
2. Trent gửi cho Alice: $\{K_B, Bob\}_K{}^{-1}{}_T$
3. Alice kiểm tra chữ ký của Trent, tạo số N_A và gửi cho Bob: $\{N_A, Alice\}_{KB}$
4. Bob giải mã, kiểm tra danh định của Alice và gửi cho Trent: *Bob, Alice*
5. Trent gửi cho Bob: $\{K_A, Alice\}_K{}^{-1}{}_T$
6. Bob kiểm tra chữ ký của Trent, tạo số N_B và gửi cho Alice: $\{N_A, N_B\}_{KA}$
7. Alice giải mã và gửi cho Bob: $\{N_B\}_{KB}$

Giao thức dùng Mã hóa công khai

- Kết quả của giao thức là Alice và Bob cùng có chung hai số nonce N_A và N_B. Khóa chung bí mật được tạo thành từ 2 số này.
- Nhưng …

  … vẫn có cách tấn công giao thức này
  - Được khám phá sau 17 năm
  - Cách tấn công: Malice lợi dụng lúc Alice muốn nói chuyện với mình để giả mạo Alice

Tấn công giao thức dùng Mã hóa công khai



Giao thức giữa Alice & Malice

Giao thức giữa Malice("Alice") & Bob

Tấn công giao thức dùng Mã hóa công khai

- Kết quả của tấn công này là:
  - Bob nghĩ rằng mình đang trao đổi 2 số nonce bí mật NA, NB với Alice nhưng thật ra là với Malice
  - Alice và Malice vẫn có cuộc nói chuyện bình thường.
- Ví dụ: Nếu Bob là một ngân hàng, Malice("Alice") gửi cho Bob một yêu cầu sau:

$\{N_A, N_B, \text{“Transfer £1B from my account to Malice's”}\}_{KB}$

Tấn công giao thức dùng Mã hóa công khai

- Cách khắc phục: kết hợp thêm chữ ký điện tử lên NA, NB
- Giao thức được sử dụng hiện nay là:
  - Gọi chung cặp khóa bí mật và khóa công khai của Alice là K_A
  - $\{M\}_K$: mã hóa M bằng khóa K
  - $[M]_K$: chữ ký điện tử lên M bằng khóa K

The Needham-Schroeder Public-key Authentication Protocol in Refined Specification

1. Alice sends to Bob : $\{[NA, Alice]_{KA}\}_{KB}$;
2. Bob sends to Alice : $\{NA, [NB]_{KB}\}_{KA}$;
3. Alice sends to Bob : $\{[NB]_{KA}\}_{KB}$.

Nội dung

1. Những khái niệm cơ bản về mã hóa
2. Mã hóa hoàn hảo
3. Kênh trao đổi khóa
4. Mô hình Dolev-Yao
5. Giao thức trao đổi khóa


Question ?