**CHUNG**

**Tính bí mật:** Bảo vệ dữ liệu không bị lộ ra ngoài một cách trái phép

**Tính toàn vẹn:** Chỉ những người dùng được ủy quyền mới được phép chỉnh sửa dữ liệu

**Tính sẵn sàng:** Đảm bảo dữ liệu luôn sẵn sàng khi những người dùng hoặc ứng dụng ủy quyền yêu cầu

**Tính chống thoái thác:** Khả năng ngăn chặn việc từ chối một hành vi đã làm

**ÔN TẬP CHI TIẾT**

**Câu 8:** Trình bày thuật toán RSA. Cho biết ưu và nhược điểm của RSA. Nêu nguyên tắc của mã hóa khóa công khai? Tại sao trong mã hóa khóa công khai không cần dùng đến kênh an toàn để truyền khóa?

**Bài làm**

**RSA (Rivest, Shamir & Adleman)** là một **hệ thống mật mã khóa công khai (PKC) được sử dụng rộng rãi**, được phát minh vào năm 1977. Nó được biết đến là rất linh hoạt, cung cấp **tính bảo mật, xác thực và chữ ký số**. Tính bảo mật của RSA dựa vào **độ khó tính toán của việc phân tích các số rất lớn** thành các thừa số nguyên tố của chúng.  
 Các bước thực hiện thuật toán RSA

Bước 1: Khởi tạo khóa

1. Chọn 2 số nguyên tố lớn p, q và tính N = p\*q sao cho

M < 2i-1 < N < 2i

1. Tính = (p - 1) \* (q - 1)
2. Tìm một số e sao cho e nguyên tố cùng nhau với

UCLN() = 1; 1 < e <

1. Tìm một số d sao cho e.d1mod (tức d là nghịch đảo của e trong phép modulo n)
2. Hủy bỏ n,p và q. Chọn cặp khóa cong khai Ku là cặp (e, N), khóa riêng KR là cặp (d, N)

Bước 2: Mã hóa bản rõ M thành bản mã C theo 2 phương pháp

1. Mã hóa bảo mật: C = E (M, KU) = Me mod N
2. Mã hóa chứng thực: C = E (M, KR) = Md mod N

Bước 3: Giải mã bản mã C thành bản rõ M

1. Mã hóa bảo mật: = D (C, KR) = C d mod N
2. Mã hóa chứng thực: = D (C, KU) = C e mod N

Ưu điểm của RSA:

1. Bảo mật toàn diện: Hỗ trợ mã hóa, xác thực và chữ ký số trong một hệ thống duy nhất
2. Quản lý khóa đơn giản: Mỗi người chỉ cần 1 cặp khóa, không cần kênh an toàn để chia sẽ khóa công khai
3. Chống chối bỏ: Khóa riêng chỉ người gửi có, nên không thể phủ nhận việc đã ký
4. Phổ biến và đáng tin cậy: Dễ tích hợp, được sử dụng rộng rãi trong thực tế

Nhược điểm của RSA:

1. Tốc độ chậm: Mã hóa và giải mã chậm hơn nhiều so với mã hóa đối xứng
2. Yêu cầu khóa dài: Khóa lớn gât tốn tài nguyên và thời gian xử lý
3. Bảo mật dựa trên giả định: Dựa vào độ khó của phân tích số nguyên, chưa có chứng minh tuyệt đối
4. Dễ sai khi triển khai: Có thể bị tấn công nếu chọn sai p, q dùng mô đun chung, số mũ nhỏ hoặc văn bản rõ đơn giản.

Nguyên tắc của mã hóa công khai:

1. Cặp khóa bất đối xứng:
   1. Mỗi người dùng một cặp khóa: khóa công khai (có thể chia sẻ) và khóa riêng (giữ bí mật)
   2. Các thao tác mã hóa/ giải mã không dùng cùng một khóa
2. Chức năng:
   1. Mã hóa: dùng khóa công khai của người nhận để mã hóa
   2. Giải mã: dùng khóa riêng của người nhận để giải mã
   3. Chữ ký số: Người gửi ký gửi bằng khóa riêng → người nhận xác minh bằng khóa công khai.
3. Cơ sở bảo mật:
   1. Dựa trên các hàm một chiều có cửa bẫy: dễ thực hiện một chiều, cực khó đảo ngược nếu không có khóa bí mật.

Tại sao không cần kênh bảo mật để truyền khóa công khai:

1. Khóa công khai không cần giữ bí mật vì:
   1. Biết khóa công khai không giúp giải mã thông điệp
   2. Không thể suy ra khóa riêng từ khóa công khai
2. Khóa công khai có thể công khai hóa, hoàn toàn, gửi qua email, website, hoặc thư mục số.
3. So với mã hóa đối xứng (SKC), PKC **loại bỏ rủi ro khi phân phối khóa bí mật** và **đơn giản hóa việc quản lý khóa**.

**Câu 9:** Khóa là gì? Trong các hệ thống mã hóa, có các loại khóa nào? Hãy liệt kê tên (tiếng anh và tiếng việt), đặc điểm chính, đóng vai trò gì trong từng loại hệ thống mã hóa. Tại sao cần giữ bí mật khóa chỉ có người gửi và người nhận biết?

**Bài làm**

Trong mật mã học, khóa là một thông số điều khiển quan trọng trong các thuật toán mã hóa và giải mã. Theo định luật **Kerckhoff**, toàn bộ độ an toàn của hệ thống mật mã **phụ thuộc vào việc giữ bí mật khóa**, không phải vào việc giấu thuật toán.

**Các loại khóa trong hệ thống mã hóa**

**1. Khóa đối xứng**

* **Tên tiếng Anh**: Symmetric Key Cryptosystem (SKC), Secret Key Cryptosystem.
* **Tên tiếng Việt**: Hệ mật mã khóa đối xứng, Hệ mật mã khóa bí mật.

**Đặc điểm chính**:

* Người gửi và người nhận dùng **chung một khóa bí mật** để mã hóa và giải mã.
* Tốc độ xử lý nhanh, nhưng phải **trao đổi khóa qua kênh an toàn**.

**Vai trò**:

* Khóa được dùng cho cả hai quá trình mã hóa và giải mã.
* Thường sử dụng trong các hệ thống cần hiệu suất cao (ví dụ: mã hóa dữ liệu lớn).

**Thách thức**:

* **Vấn đề phân phối khóa**: khó chia sẻ khóa an toàn nếu không có kênh tin cậy.
* Cần **n(n−1)/2 khóa** cho n người dùng → khó mở rộng và quản lý.

**2. Khóa bất đối xứng**

* **Tên tiếng Anh**: Public Key Cryptosystem (PKC), Asymmetric Key Cryptosystem.
* **Tên tiếng Việt**: Hệ mật mã khóa công khai, Hệ mật mã khóa phi đối xứng.

**Đặc điểm chính**:

* Mỗi người có một **cặp khóa**:
  + **Khóa công khai**: chia sẻ rộng rãi để mã hóa và xác minh chữ ký.
  + **Khóa riêng**: giữ bí mật tuyệt đối, dùng để giải mã và ký.

**Vai trò**:

* Giải quyết **vấn đề phân phối khóa**.
* Hỗ trợ **chữ ký số**, **xác thực**, và **chống chối bỏ**.

**Ưu điểm**:

* Không cần kênh an toàn để chia sẻ khóa công khai.
* Mỗi người chỉ cần quản lý một cặp khóa → đơn giản hóa hệ thống.

**Tại sao cần giữ bí mật khóa?**

1. **Cơ sở của bảo mật**: Nếu khóa bị lộ, toàn bộ hệ thống mất an toàn – bất kể thuật toán có mạnh đến đâu.
2. **Bảo vệ tính bí mật**: Người ngoài có khóa → có thể giải mã thông tin.
3. **Đảm bảo toàn vẹn và xác thực**: Trong hệ PKC, nếu lộ khóa riêng, kẻ tấn công có thể giả mạo chữ ký số.
4. **Ngăn chặn các cuộc tấn công**: Brute-force, dictionary attack, replay attack...

**Câu 10:** Mã hóa bất đối xứng dùng 2 khóa khác nhau cho 2 quá trình mã hóa và giải mã. Trình bày (có giải thích) việc dùng phương pháp mã hóa bất đối xứng để giải quyết bài toán a. Bảo mật dữ liệu.

**Bài làm**

Trong **mã hóa bất đối xứng (Public Key Cryptosystem – PKC)**, mỗi người dùng tạo một **cặp khóa** gồm:

* **Khóa công khai (public key)**: chia sẻ rộng rãi.
* **Khóa bí mật (private key)**: chỉ chủ sở hữu biết và giữ bí mật.

Cơ chế này sử dụng **2 khóa khác nhau** cho mã hóa và giải mã, giúp giải quyết:

**🔒 a. Bảo mật dữ liệu (Data Confidentiality)**

**Cách hoạt động:**

* Người gửi (Bob) dùng **khóa công khai của người nhận (Alice)** để mã hóa thông điệp.
* Chỉ người nhận (Alice) có thể dùng **khóa bí mật của mình** để giải mã.

**Giải thích:**

* Kẻ tấn công không thể giải mã nếu không có khóa bí mật của Alice.
* Toàn bộ tính bảo mật phụ thuộc vào việc **giữ bí mật khóa riêng**.

**Ai tạo khóa?**

* **Người nhận (Alice)** tạo ra cặp khóa công khai – bí mật của mình.

**✅ b. Chứng thực nguồn gốc thông điệp (Authentication)**

**Cách hoạt động:**

* Người gửi (Alice) dùng **khóa bí mật của chính mình** để **ký thông điệp** (thường là mã băm của thông điệp).
* Người nhận (Bob) dùng **khóa công khai của Alice** để xác minh chữ ký.

**Giải thích:**

* Vì chỉ Alice có khóa bí mật → chỉ cô ấy có thể tạo ra chữ ký hợp lệ.
* Bob kiểm tra và xác minh được rằng thông điệp thực sự từ Alice và **không bị sửa đổi**.
* Đồng thời, điều này cũng đảm bảo **tính chống chối bỏ** (non-repudiation).

**Ai tạo khóa?**

* **Người gửi (Alice)** tạo ra cặp khóa công khai – bí mật của mình.

**Câu 11:** Hệ mã hóa đối xứng là gì? vẽ mô hình cơ bản và cho biết các thành phần cơ bản của mã hóa đối xứng? Trình bày ưu điểm và hạn chế của hệ mã đối xứng

**🔐 1. Định nghĩa hệ mã hóa đối xứng**

**Mã hóa đối xứng** (Symmetric Key Cryptosystem) là hệ mã trong đó **cùng một khóa bí mật được sử dụng cho cả quá trình mã hóa và giải mã**. Khóa này chỉ được biết bởi hai bên: người gửi và người nhận. Vai trò giữa hai bên là đối xứng, nghĩa là có thể thay phiên nhau gửi và nhận thông điệp bằng cùng một khóa.

**⚙️ 2. Các thành phần cơ bản của hệ mã hóa đối xứng**

1. M (Plaintext): Thông điệp gốc – bản rõ.
2. C (Ciphertext): Bản mã – thông điệp đã được mã hóa.
3. Khóa bí mật (KS): Được sử dụng cả để mã hóa và giải mã.
4. Thuật toán mã hóa: Chuyển bản rõ M thành bản mã C.
5. Thuật toán giải mã: Chuyển bản mã C trở lại thành bản rõ M.
6. Kênh truyền: Nơi truyền dữ liệu (C) từ người gửi đến người nhận.

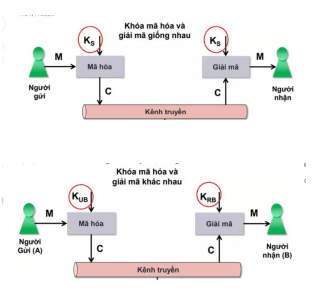
**✅ 3. Ưu điểm của mã hóa đối xứng**

* **Tốc độ cao**: Mã hóa và giải mã nhanh, hiệu quả khi xử lý lượng lớn dữ liệu.
* **Tiết kiệm tài nguyên**: Yêu cầu tính toán thấp, phù hợp với thiết bị giới hạn hiệu năng.
* **Phổ biến**: Dễ triển khai trong các hệ thống nội bộ hoặc môi trường có quản lý khóa hiệu quả.

**⚠️ 4. Hạn chế của mã hóa đối xứng**

* **Quản lý khóa phức tạp**:
  + Trong hệ thống có n người dùng, cần tới **n(n-1)/2 khóa riêng biệt**, gây khó khăn trong lưu trữ và chia sẻ.
* **Thiết lập khóa an toàn khó khăn**:
  + Nếu hai bên chưa từng trao đổi trước đó, **khó có cách an toàn để chia sẻ khóa** qua kênh công cộng.
* **Dễ bị phân tích mật mã nếu dùng sai cách**:
  + Dùng một khóa cho quá nhiều bản mã sẽ tạo điều kiện cho kẻ tấn công thực hiện phân tích.
* **Thiếu xác thực và chống chối bỏ**:
  + Vì cả hai bên đều biết khóa nên **không thể chứng minh ai là người gửi**, gây khó khăn trong các hệ thống cần chữ ký số hoặc xác minh trách nhiệm.

**Câu 12:** Vẽ mô hình hệ mã hóa đối xứng và hệ mã hóa bất đối xứng? So sánh hệ mã đối xứng và hệ mã bất đối xứng



**🔐 So sánh hệ mã đối xứng và hệ mã bất đối xứng**

**1. Hệ mã đối xứng (Symmetric-key Encryption)**

* Sử dụng **cùng một thuật toán và cùng một khóa** để thực hiện **mã hóa và giải mã**.
* Người gửi (Sender) và người nhận (Receiver) **phải chia sẻ chung thuật toán và khóa**.
* **Khóa phải được giữ bí mật tuyệt đối**, vì nếu bị lộ thì toàn bộ dữ liệu có thể bị giải mã.
* Không thể (hoặc rất khó) giải mã một thông điệp nếu **thiếu khóa bí mật**, ngay cả khi có sẵn thông tin khác.
* **Chỉ khi biết thuật toán + có mẫu ciphertext đủ nhiều**, mới có thể cố gắng xác định lại khóa (phân tích mật mã).

**2. Hệ mã bất đối xứng (Public-key Encryption)**

* Sử dụng **một thuật toán duy nhất** nhưng hoạt động với **hai khóa khác nhau**:  
  → **Khóa công khai** dùng để mã hóa và  
  → **Khóa bí mật** dùng để giải mã.
* Người gửi và người nhận **không cần chia sẻ cùng một khóa**. Mỗi người giữ **một cặp khóa riêng**.
* **Chỉ một trong hai khóa (thường là khóa bí mật)** cần được bảo vệ.
* Giống như hệ đối xứng: không thể hoặc rất khó giải mã nếu thiếu khóa bí mật.
* Cần **thuật toán + một trong hai khóa + mẫu ciphertext đủ lớn** mới có thể xác định lại khóa còn lại (nếu có thể).

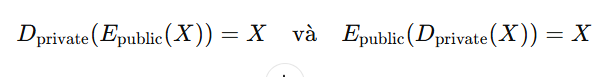
**Câu 13:** Trong mã hóa khóa công khai, khóa riêng và khóa công khai có phải là 2 khóa tùy ý, không liên quan? Nếu có liên quan, tại sao không thể tính khóa riêng từ khóa công khai? Tại sao trong hệ mã RSA nếu biết khóa công khai (n,e) thì rất khó tìm khóa bí mật (n,d)

**Bài làm**

**1. 🔗 Khóa công khai và khóa bí mật có liên quan không?**

→ **Có liên quan chặt chẽ về mặt toán học**, không phải hai khóa tùy ý.

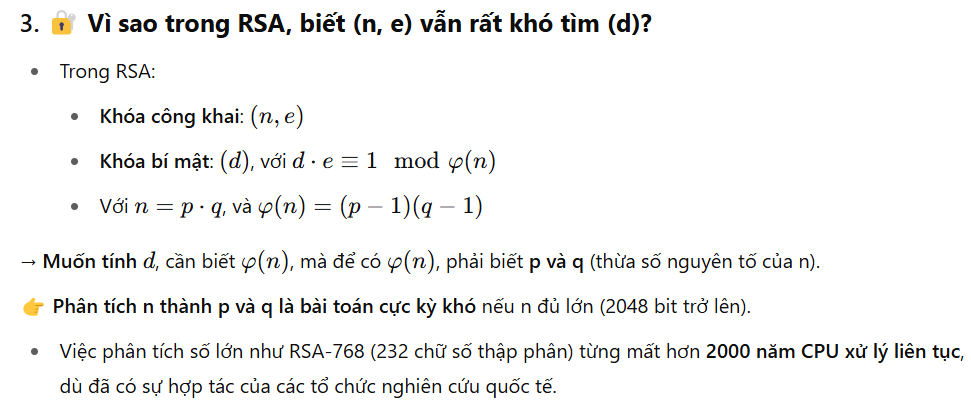
* Mỗi người dùng tạo **một cặp khóa**: khóa công khai (dùng để mã hóa/xác minh) và khóa bí mật (dùng để giải mã/ký).
* Cặp khóa được thiết kế sao cho:



**2. ❓ Vì sao không thể tính khóa bí mật từ khóa công khai?**

→ Vì mã hóa khóa công khai sử dụng **hàm một chiều có cửa bẫy (trapdoor one-way function)**:

* **Hàm một chiều**: dễ tính theo một chiều, nhưng **rất khó đảo ngược** nếu không biết thông tin bí mật.
* **Cửa bẫy**: nếu biết khóa bí mật, việc giải mã rất dễ; nếu không, việc tìm lại khóa là **gần như không thể thực hiện** bằng công cụ tính toán hiện nay.



**Câu 14:** Khóa phiên (Session Key) là gì? Khóa phiên khác khóa bí mật chia sẻ (secret key) như thế nào? (Vẽ mô hình KDC (Key Distribution Center))

**Bài làm**

Khóa phiên là **khóa đối xứng tạm thời**, chỉ được sử dụng trong **một phiên giao tiếp duy nhất** giữa hai bên (ví dụ: người dùng và máy chủ). Sau khi phiên kết thúc, khóa này sẽ bị hủy bỏ.

**Sự khác nhau giữa Khóa phiên (Session Key) và Khóa bí mật chia sẻ (Secret Key)**

**Khóa phiên (session key)** và **khóa bí mật chia sẻ (secret key)** đều là những thành phần quan trọng trong hệ thống mật mã đối xứng, tuy nhiên chúng khác nhau ở mục đích sử dụng, thời gian tồn tại và cách thiết lập.

**1. Thời gian tồn tại:**

* **Khóa phiên** là khóa tạm thời, chỉ tồn tại trong suốt một phiên giao tiếp cụ thể giữa hai bên. Khi phiên kết thúc, khóa này sẽ bị hủy bỏ và không tái sử dụng.
* **Khóa bí mật chia sẻ** là khóa lâu dài, được thiết lập trước giữa hai bên và có thể sử dụng cho nhiều phiên giao tiếp.

**2. Mục đích sử dụng:**

* **Khóa phiên** được sử dụng để mã hóa dữ liệu trong một phiên cụ thể nhằm đảm bảo tính bảo mật và hiệu quả xử lý. Việc thay đổi khóa thường xuyên giúp giảm thiểu rủi ro nếu khóa bị lộ.
* **Khóa bí mật chia sẻ** thường được dùng để xác thực lẫn nhau giữa hai bên, hoặc để thiết lập các khóa phiên một cách an toàn thông qua các giao thức trao đổi khóa.

**3. Cách thiết lập:**

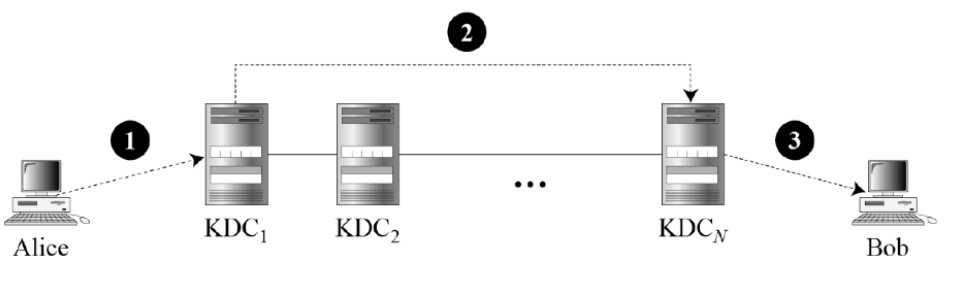
* **Khóa phiên** thường được sinh ra ngẫu nhiên trong quá trình khởi tạo phiên và được trao đổi qua các giao thức an toàn như Diffie-Hellman, RSA hoặc thông qua một bên thứ ba đáng tin cậy như trong Kerberos.
* **Khóa bí mật chia sẻ** phải được thiết lập từ trước, thông qua một kênh an toàn (có thể là trao tay hoặc sử dụng hạ tầng khóa bí mật chung) và được giữ bí mật tuyệt đối trong suốt thời gian sử dụng.

**4. Phạm vi ảnh hưởng nếu bị lộ:**

* Nếu **khóa phiên** bị lộ, kẻ tấn công chỉ có thể giải mã dữ liệu của phiên tương ứng, không ảnh hưởng đến các phiên khác.
* Nếu **khóa bí mật chia sẻ** bị lộ, tất cả các phiên và dữ liệu đã sử dụng khóa đó đều có nguy cơ bị xâm phạm.

**5. Loại khóa:**

* **Khóa phiên** luôn là khóa đối xứng, vì mục tiêu chính là tối ưu hiệu suất trong mã hóa và giải mã dữ liệu.
* **Khóa bí mật chia sẻ** có thể là khóa đối xứng (trong hệ mật mã đối xứng) hoặc khóa bí mật (private key) trong hệ mật mã công khai (PKC).



**Câu 15:** Thế nào là tấn công Man-in-the-middle. Nêu (vẽ mô hình) và giải thích một giao thức/cơ chế mà có thể bị tấn công này tấn công.

**Bài làm**

**Tấn công Man-in-the-Middle (kẻ ở giữa)** là một loại **tấn công chủ động** trong đó **kẻ tấn công chèn mình vào giữa quá trình giao tiếp** của hai bên (ví dụ: Alice và Bob), khiến họ **tưởng rằng đang giao tiếp trực tiếp với nhau**, trong khi toàn bộ dữ liệu đều bị **kẻ tấn công kiểm soát**. **Mục tiêu của kẻ tấn công là:**

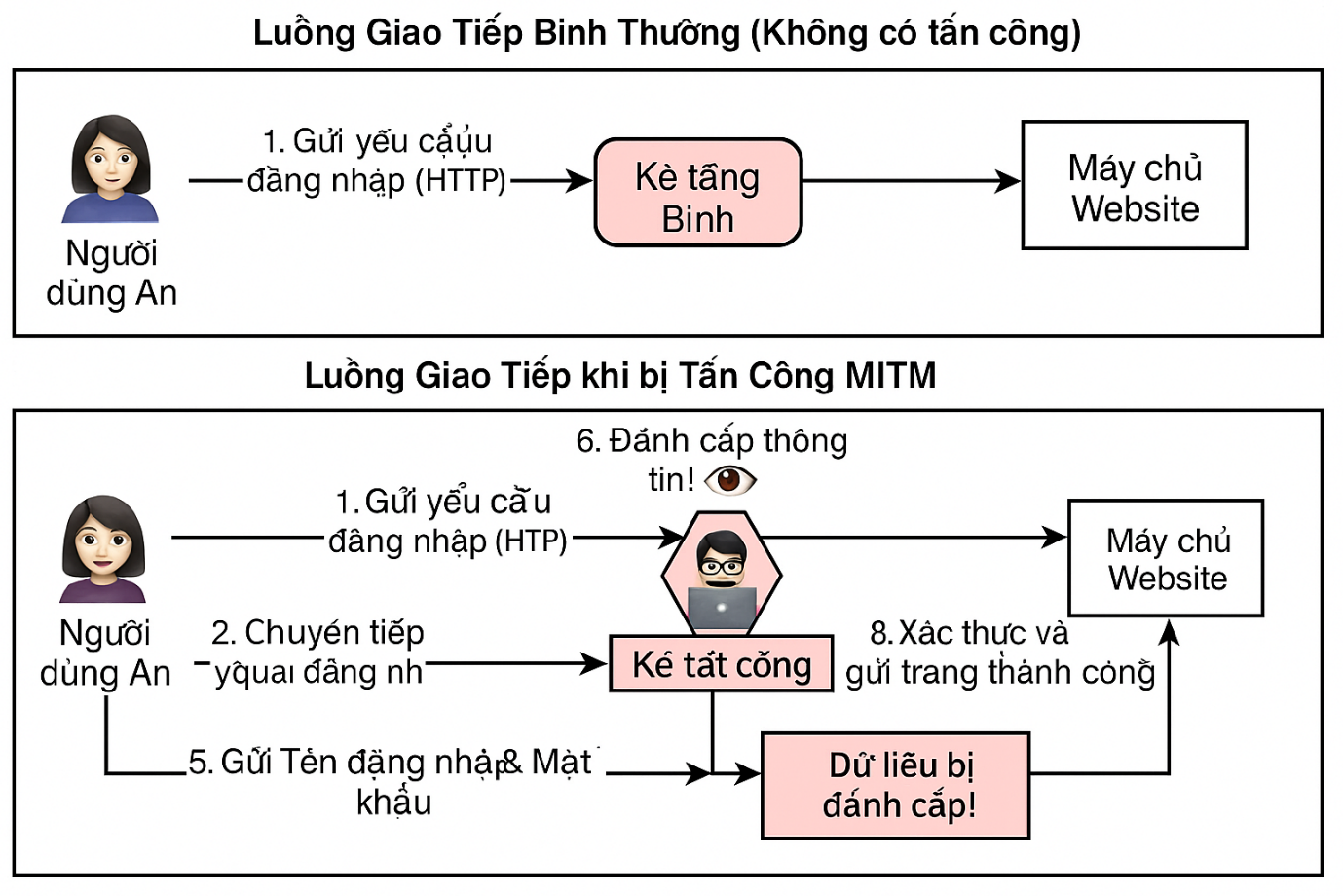
* **Nghe lén:** Đánh cắp thông tin nhạy cảm như tài khoản ngân hàng, mật khẩu, số thẻ tín dụng.
* **Thao túng dữ liệu:** Sửa đổi nội dung trao đổi mà hai bên không hề hay biết (ví dụ: thay đổi số tài khoản nhận tiền trong một giao dịch).
* **Chiếm quyền phiên (Session Hijacking):** Đánh cắp cookie phiên để giả mạo danh tính người dùng và truy cập vào các tài khoản của họ.

Giao Thức Bị Tấn Công: HTTP Trên Mạng Wi-Fi Công Cộng

Một trong những kịch bản phổ biến và dễ bị tấn công MITM nhất là khi người dùng truy cập một trang web sử dụng **giao thức HTTP** (không được mã hóa) qua một mạng Wi-Fi công cộng không an toàn (quán cà phê, sân bay, khách sạn).

Giả sử người dùng tên **An** đang ở quán cà phê và kết nối vào mạng Wi-Fi miễn phí. Một kẻ tấn công tên **Bình** cũng đang ở trong cùng mạng đó.

1. **Thiết lập bẫy:** Bình sử dụng các công cụ chuyên dụng (như ARP spoofing hoặc tạo một điểm truy cập Wi-Fi giả mạo) để điều hướng toàn bộ lưu lượng mạng của An đi qua máy tính của mình trước khi ra ngoài Internet.
2. **An truy cập trang web:** An mở trình duyệt và truy cập vào một trang web không an toàn, ví dụ http://muasam-online.com (sử dụng giao thức HTTP). Yêu cầu truy cập của An bị máy của Bình chặn lại **(Bước 1)**.
3. **Bình chuyển tiếp yêu cầu:** Bình nhận được yêu cầu và chuyển tiếp nó đến máy chủ thực của muasam-online.com **(Bước 2)**. Đối với máy chủ, yêu cầu này trông như thể nó đến trực tiếp từ An.
4. **Máy chủ phản hồi:** Máy chủ gửi lại trang đăng nhập. Phản hồi này lại đi qua máy của Bình trước khi đến được máy của An **(Bước 3 & 4)**. Lúc này, An vẫn thấy trang web hiện ra bình thường và không hề nghi ngờ.
5. **An nhập thông tin:** An nhập tên đăng nhập và mật khẩu của mình vào trang web. Vì trang web sử dụng HTTP, thông tin này được gửi đi dưới dạng **văn bản thuần (plaintext)**, không mã hóa **(Bước 5)**.
6. **Đánh cắp dữ liệu:** Khi gói tin chứa tên đăng nhập và mật khẩu của An đi qua máy của Bình, Bình có thể dễ dàng đọc và lưu lại thông tin này **(Bước 6)**.
7. **Hoàn tất giao dịch:** Để An không nghi ngờ, Bình chuyển tiếp thông tin đăng nhập của An đến máy chủ. Máy chủ xác thực thông tin là đúng và cho phép An đăng nhập thành công. Toàn bộ quá trình sau đó diễn ra bình thường trong khi An không hề hay biết mật khẩu của mình đã bị đánh cắp **(Bước 7, 8, 9)**.



**Câu 16:** Trình bày giao thức trao đổi khóa Diffie-Hellman. Nêu ưu điểm và nhược điểm của giao thức trao đổi khóa Diffie-Helman.

**Bài làm**

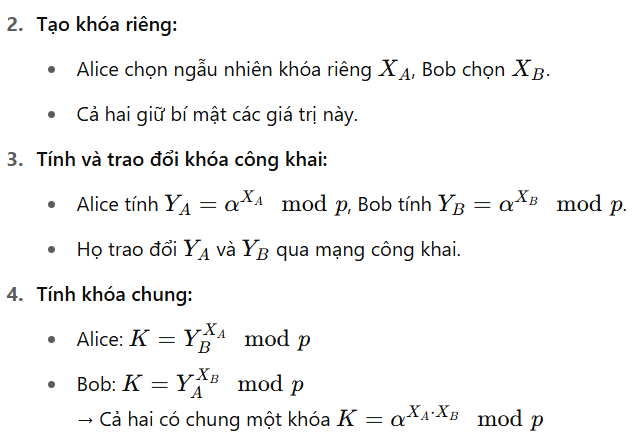
**Giao thức trao đổi khóa Diffie-Hellman**

**✅ Mục tiêu:**

Cho phép hai bên (ví dụ: Alice và Bob) **thiết lập một khóa bí mật chung** qua kênh truyền **không an toàn**, **không cần bên thứ ba tin cậy**.

**🔁 Nguyên lý hoạt động:**

1. **Thỏa thuận tham số công khai:**
   * Chọn một số nguyên tố lớn ppp và một số nguyên sinh α\alphaα.
   * Các giá trị này được công khai.



**✅ Ưu điểm của Diffie-Hellman:**

* **Thiết lập khóa an toàn qua mạng công khai**, không cần chia sẻ khóa trước.
* **Không cần máy chủ cấp phát khóa** (KDC), giảm phụ thuộc bên thứ ba.
* **Ứng dụng rộng rãi** trong các giao thức như TLS/SSL, IPsec.

**❌ Nhược điểm của Diffie-Hellman:**

* **Dễ bị tấn công Man-in-the-Middle (MitM):**
  + Kẻ tấn công (Mallory) chèn public key của mình thay vì của đối phương.
  + Kết quả: Alice và Bob tưởng đang trao đổi với nhau nhưng thực chất đang nói chuyện với Mallory.
* **Không có cơ chế xác thực**: Giao thức cơ bản không kiểm tra được danh tính người gửi → dễ bị giả mạo.

🛡 **Khắc phục**: Cần kết hợp với **chữ ký số** hoặc **chứng chỉ số (CA)** để xác thực public key.

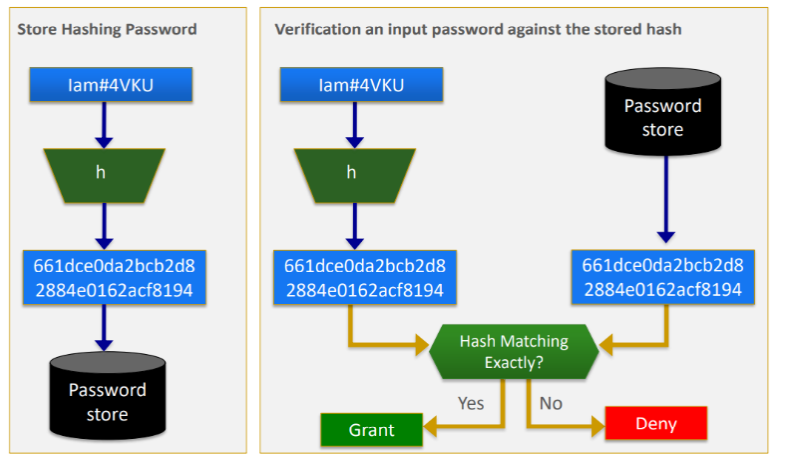
**Câu 17:**Hàm băm là gì? Nêu và giải thích ứng dụng hàm băm trong việc lưu trữ mật khẩu (Vẽ mô hình).

**Bài làm**

Hàm băm đóng vai trò quan trọng trong việc bảo mật mật khẩu người dùng khi lưu trữ trong hệ thống. Thay vì lưu trữ mật khẩu gốc (plaintext), hệ thống sẽ lưu trữ giá trị băm của mật khẩu đó. Điều này giúp bảo vệ mật khẩu ngay cả khi cơ sở dữ liệu bị xâm nhập.

**Mô Hình Lưu Trữ Mật Khẩu Bằng Hàm Băm**

Dưới đây là mô hình minh họa quá trình lưu trữ và xác thực mật khẩu sử dụng hàm băm:



**1. Lưu Trữ Mật Khẩu (Store Hashing Password)**

Khi một người dùng tạo tài khoản hoặc thay đổi mật khẩu:

1. **Nhập Mật Khẩu:** Người dùng cung cấp mật khẩu của họ (ví dụ: "Iam#4VKU").
2. **Băm Mật Khẩu:** Mật khẩu này sẽ được đưa qua một **hàm băm (h)**.
3. **Lưu Giá Trị Băm:** Kết quả của hàm băm là một chuỗi ký tự có độ dài cố định (ví dụ: "661dce0da2bcb2d82884e0162acf8194"). Giá trị băm này sẽ được lưu trữ trong cơ sở dữ liệu **(Password store)**, thay vì mật khẩu gốc.

**2. Xác Thực Mật Khẩu (Verification an input password against the stored hash)**

Khi người dùng cố gắng đăng nhập:

1. **Nhập Mật Khẩu:** Người dùng nhập mật khẩu của họ (ví dụ: "Iam#4VKU").
2. **Băm Mật Khẩu Nhập:** Mật khẩu vừa nhập lại được đưa qua **cùng hàm băm (h)** đã sử dụng khi lưu trữ.
3. **So Sánh Giá Trị Băm:** Giá trị băm mới được tạo ra từ mật khẩu nhập vào sẽ được so sánh chính xác với giá trị băm đã được lưu trữ trong cơ sở dữ liệu tương ứng với tài khoản đó.
4. **Kết Quả:**
   * **Khớp (Yes):** Nếu hai giá trị băm hoàn toàn giống nhau, hệ thống xác nhận mật khẩu đúng và **cấp quyền truy cập (Grant)**.
   * **Không Khớp (No):** Nếu hai giá trị băm khác nhau, hệ thống từ chối quyền truy cập **(Deny)**.

**Giải Thích Ứng Dụng**

* **Bảo mật:** Ngay cả khi kẻ tấn công có được quyền truy cập vào cơ sở dữ liệu, họ chỉ thấy các giá trị băm đã được mã hóa, chứ không thấy mật khẩu gốc của người dùng. Việc đảo ngược hàm băm để lấy lại mật khẩu gốc là cực kỳ khó khăn về mặt tính toán (tính một chiều của hàm băm).
* **Ngăn chặn lộ mật khẩu:** Vì hệ thống không bao giờ lưu trữ mật khẩu gốc, nguy cơ mật khẩu bị lộ do lỗi hệ thống hoặc nhân viên nội bộ cũng được giảm thiểu đáng kể.
* **Tính toàn vẹn:** Nếu mật khẩu gốc bị thay đổi, giá trị băm tương ứng cũng sẽ thay đổi hoàn toàn. Điều này đảm bảo tính toàn vẹn của thông tin xác thực.

**Câu 18:** Hàm băm là gì? Nêu và giải thích ứng dụng hàm băm trong chữ ký điện tử (Vẽ mô hình).

**Bài làm**

**Hàm băm** là một hàm toán học nhận đầu vào là một chuỗi dữ liệu bất kỳ (X) và **trả về một chuỗi đầu ra có độ dài cố định** (h(X)), gọi là **giá trị băm (hash value hoặc digest)**.

* Ví dụ: một văn bản dài hàng trăm KB có thể được băm thành một chuỗi chỉ dài 128 hoặc 256 bit.
* Hàm băm được sử dụng rộng rãi trong **chữ ký số**, **xác thực dữ liệu**, và các **ứng dụng bảo mật**.

Các đặc tính quan trọng của một hàm băm an toàn bao gồm:

* **Tính xác định:** Cùng một dữ liệu đầu vào luôn tạo ra cùng một giá trị băm.
* **Tính một chiều:** Từ giá trị băm, việc tìm lại dữ liệu gốc là cực kỳ khó khăn về mặt tính toán.
* **Tính kháng xung đột:** Rất khó để tìm thấy hai dữ liệu đầu vào khác nhau mà lại tạo ra cùng một giá trị băm.
* **Hiệu quả:** Quá trình tạo giá trị băm phải diễn ra nhanh chóng.

**Ứng Dụng Của Hàm Băm Trong Chữ Ký Điện Tử**

Trong chữ ký điện tử, hàm băm được ứng dụng để đảm bảo **tính toàn vẹn** và **tính xác thực** của dữ liệu một cách hiệu quả. Thay vì ký (mã hóa) toàn bộ một tài liệu lớn vốn rất chậm, người ta chỉ ký lên "dấu vân tay số" nhỏ gọn của nó.

Quá trình này được giải thích qua mô hình dưới đây.

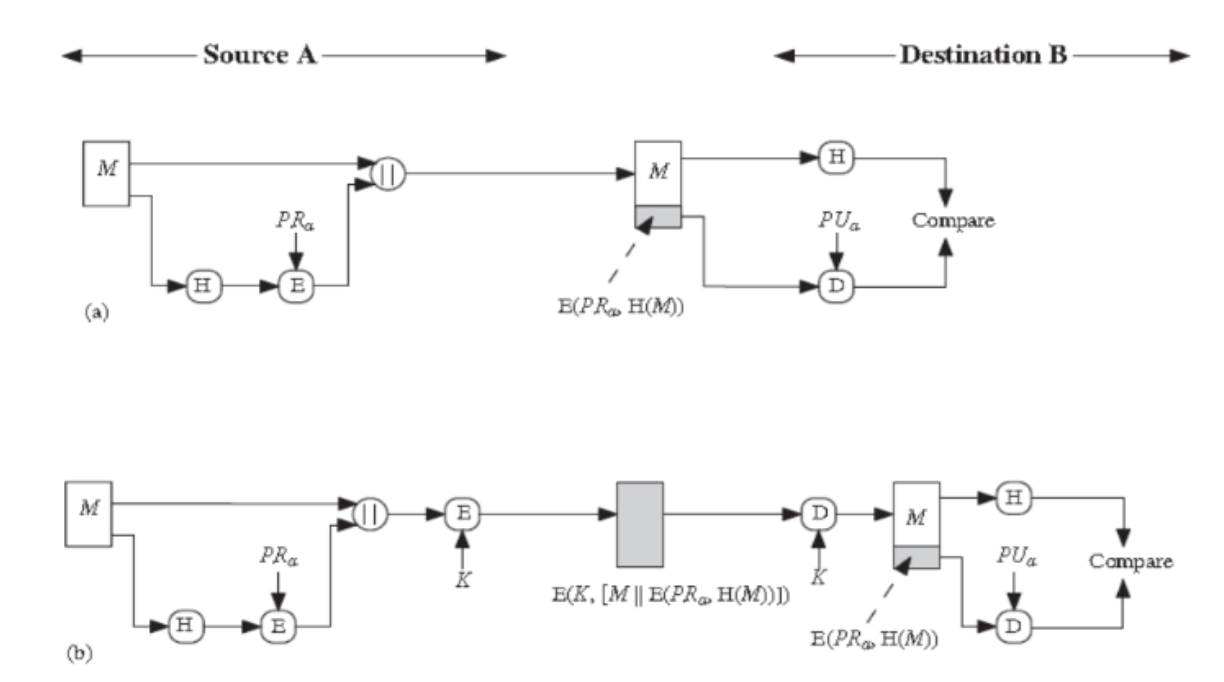
**Giải thích mô hình**

**1. Phía Người Gửi (Tạo chữ ký)**

* **Bước 1: Tạo giá trị băm.** Thông điệp gốc **(M)** được đưa qua một **hàm băm (H)** để tạo ra một "dấu vân tay số" duy nhất là **H(M)**.
* **Bước 2: Tạo chữ ký.** "Dấu vân tay" **H(M)** này được mã hóa **(E)** bằng **khóa bí mật (PRa)** của người gửi. Kết quả của quá trình này chính là **Chữ ký điện tử (S)**.
* **Bước 3: Gửi đi.** Chữ ký điện tử được đính kèm vào thông điệp gốc và gửi đi.

**2. Phía Người Nhận (Xác thực chữ ký)**

* **Bước 4: Tính toán lại giá trị băm.** Người nhận lấy thông điệp nhận được **(M\_rec)** và dùng đúng hàm băm đó **(H2)** để tự tính toán ra một giá trị băm mới **(HM\_rec)**.
* **Bước 5: Giải mã chữ ký.** Đồng thời, họ dùng **khóa công khai (PUa)** của người gửi để giải mã **(D)** chữ ký nhận được. Kết quả giải mã chính là giá trị băm gốc **(HM\_orig)** mà người gửi đã tạo ra.
* **Bước 6: So sánh.** Hệ thống so sánh hai giá trị băm: giá trị mới tính toán **(HM\_rec)** và giá trị gốc được giải mã **(HM\_orig)**.
  + Nếu chúng **trùng khớp**, chữ ký hợp lệ. Điều này chứng minh rằng:
    - **Tính toàn vẹn:** Dữ liệu không hề bị thay đổi trên đường truyền.
    - **Tính xác thực:** Dữ liệu chắc chắn được ký bởi người sở hữu khóa bí mật tương ứng.
  + Nếu chúng **không khớp**, có nghĩa là dữ liệu đã bị sửa đổi hoặc chữ ký là giả mạo.



**Câu 19:** Hàm băm là gì? Trình bày và giải thích các tính chất của hàm băm?

**Bài làm**

**Hàm băm** là một hàm toán học nhận đầu vào là một chuỗi dữ liệu bất kỳ (X) và **trả về một chuỗi đầu ra có độ dài cố định** (h(X)), gọi là **giá trị băm (hash value hoặc digest)**.

* Ví dụ: một văn bản dài hàng trăm KB có thể được băm thành một chuỗi chỉ dài 128 hoặc 256 bit.
* Hàm băm được sử dụng rộng rãi trong **chữ ký số**, **xác thực dữ liệu**, và các **ứng dụng bảo mật**.

**✅ 2. Các tính chất quan trọng của hàm băm mật mã**

**🔹 a. Kích thước đầu ra cố định (Fixed Output Length)**

* Dù đầu vào có độ dài bất kỳ, đầu ra luôn có độ dài cố định.
* Giúp tiêu chuẩn hóa việc xử lý, so sánh, và ký dữ liệu.

**🔹 b. Tính một chiều (One-Way Property)**

* Dễ tính h(X) từ X, nhưng **gần như không thể tìm X khi chỉ biết h(X)**.
* Đảm bảo dữ liệu không bị đảo ngược từ giá trị băm → giúp bảo mật.

**🔹 c. Tính chống đụng độ (Collision Resistance)**

* Rất khó để tìm **hai đầu vào khác nhau X ≠ X'** sao cho h(X) = h(X').
* Dù va chạm là lý thuyết có thể xảy ra (do đầu ra có độ dài giới hạn), nhưng hàm băm tốt khiến việc này **gần như bất khả thi**.

**✍️ 4. Vai trò trong bảo mật**

* Hàm băm giúp **ký các thông điệp dài** bằng cách chỉ ký trên giá trị băm → tiết kiệm tài nguyên, tăng hiệu quả.
* Đảm bảo **toàn vẹn nội dung**, phát hiện thay đổi bất hợp pháp.

**Câu 20:** Chữ ký số là gì? Trình bày và giải thích quá trình tạo chữ ký số và thẩm tra chữ ký số

**Bài làm**

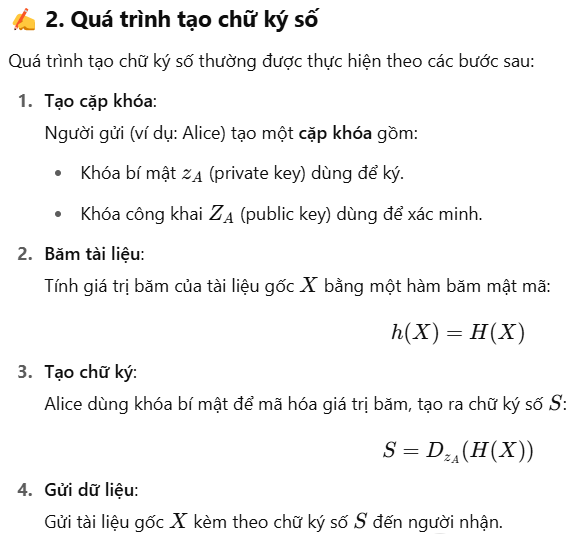
**🔐 1. Chữ ký số là gì?**

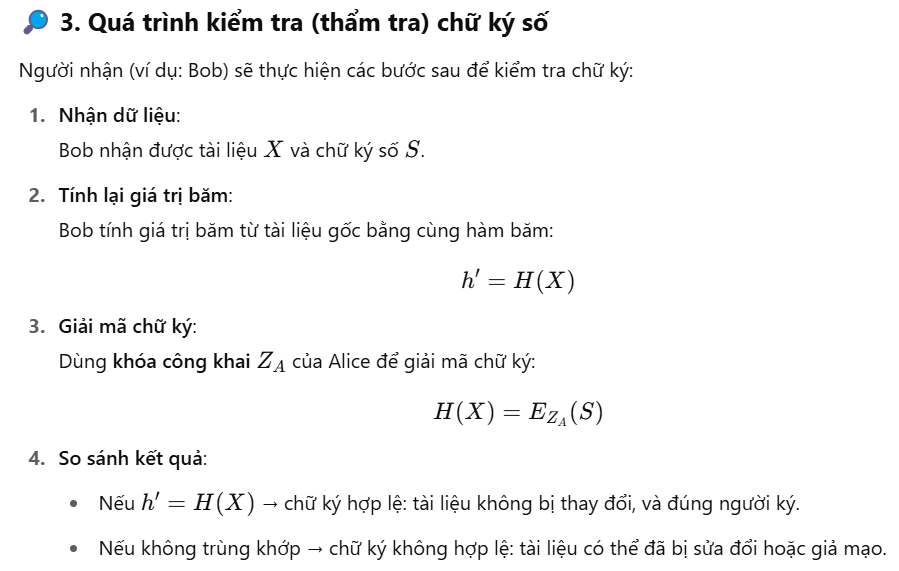
Chữ ký số (Digital Signature) là một **chuỗi nhị phân đặc biệt** được tạo ra bằng cách dùng **khóa bí mật** để ký lên bản băm của tài liệu gốc. Nó có tác dụng xác minh:

* **Nguồn gốc** (ai gửi),
* **Toàn vẹn** (nội dung không bị sửa),
* **Chống chối bỏ** (không thể phủ nhận đã ký).

Chữ ký số hoạt động dựa trên hai thành phần chính:

* **Mã hóa khóa công khai (PKC)**
* **Hàm băm mật mã**

****

****