



## Bài 5. Xác thực danh tính

ONE LOVE. ONE FUTURE.

1

1

### Nội dung

- Khái niệm chung
- Xác thực dựa trên mật khẩu
- Các giao thức xác thực dựa trên mật khẩu
- Giao thức zero-knowledge
- Giới thiệu một số phương pháp xác thực khác



ĐẠI HỌC BÁCH KHOA HÀ NỘI  
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

2

2

## Xác thực danh tính là gì?

- Danh tính (Identifier) ~ Định danh
- Xác thực danh tính: Ai đang truy cập/trao đổi dữ liệu?  
→ Khi có truy cập/trao đổi dữ liệu, hệ thống cần xác minh được liên kết giữa chủ thể ở thế giới thực với danh tính truy cập
- Một số yếu tố để xác thực:
  - Yếu tố tri thức (What the entity knows)
  - Yếu tố tài sản (What the entity has)
  - Yếu tố sinh trắc (What the entity is)
  - Yếu tố vị trí (Where the entity is)...
- Xác thực đa yếu tố: sử dụng nhiều hơn 1 yếu tố xác thực



ĐẠI HỌC BÁCH KHOA HÀ NỘI  
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

3

3

## 1. Xác thực bằng mật khẩu

ONE LOVE. ONE FUTURE.

4

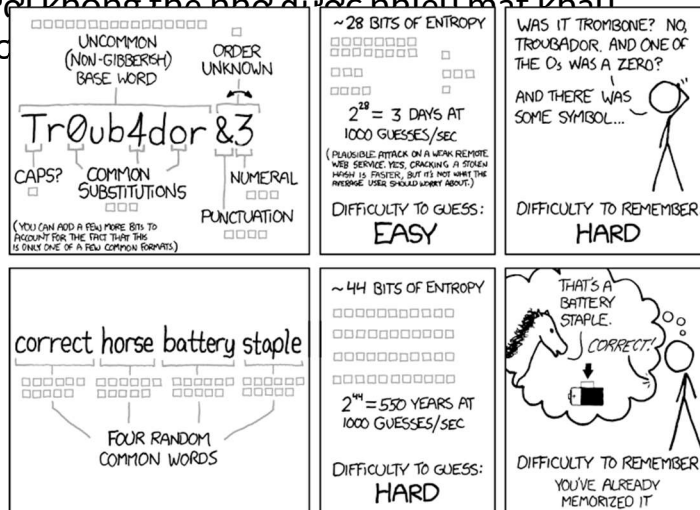
4

## Xác thực bằng mật khẩu

- Mật khẩu: một chuỗi ký tự hoặc một nhóm từ được sử dụng để xác thực danh tính của thực thể nào đó
  - Thực thể(Entity) cần xác thực (**người dùng**, thiết bị, ứng dụng...)
  - Người thẩm tra(Verifier): kiểm tra tính hợp lệ của mật khẩu
- Các lỗi hổng làm giảm sự tin cậy:
  - Lưu trữ mật khẩu trong CSDL không an toàn
  - Truyền mật khẩu trên kênh không an toàn
  - Người dùng không cẩn trọng

## Người dùng có thể bị đánh cắp tài khoản như thế nào?

- Con người không thể nhớ được nhiều mật khẩu được cho



## Người dùng có thể bị đánh cắp tài khoản như thế nào?

- Vì vậy, người dùng thường dùng lại mật khẩu cho các tài khoản khác nhau
  - Với hy vọng sẽ không xảy ra điều gì tồi tệ
- Khi mật khẩu của 1 tài khoản nào đó bị lộ?
  - Kẻ tấn công có mật khẩu của người dùng
  - Và đăng nhập vào các tài khoản khác
- Thực tế: Hacker đã thử tấn công đánh cắp các tài khoản vận hành mạng lưới điện của Mỹ theo hướng tiếp cận này
  - Gửi email giả mạo chia sẻ tài liệu Dropbox
  - Tấn công vào website có yêu cầu xác thực người dùng

## Không đổ lỗi cho người dùng

- Thông thường, chúng ta thường đổ lỗi cho người dùng khi họ sơ ý bị kẻ tấn công khai thác
- Chúng ta cần xây dựng hệ thống có khả năng hỗ trợ người dùng không hành động sai
- Ví dụ: thư giả mạo (phishing email)

2020 CẬP NHẬT MỚI NHẤT (KHÔNG IGNOR

Webmail Service Center <it-arts@sun.unipune.ac.in>  
To: -

--

Chúng tôi đang thực hiện một cuộc khảo sát xác minh thư rác và cập nhật.  
Điều quan trọng là bạn xác minh tài khoản của mình để giúp chúng

[Xác minh tài khoản của bạn ngay.](#)

Nếu xác minh không nhận được từ bạn trong 48 giờ tới, tài khoản phục vụ sẽ bị khóa. Khi bạn có thể truy cập lại tài khoản của mình

Cảm ơn Admin! © 2020 Tất cả các quyền

--  
This message has been scanned for viruses and dangerous content, and is believed to be clean.



## Giải pháp cho người dùng

- Phần mềm quản lý mật khẩu
  - Ví dụ: StickyPassword Free, RoboForm,...
  - Có thể tạo ra các mật khẩu “mạnh” và quản lý tài khoản sử dụng mật khẩu này
  - Người dùng chỉ cần nhớ 1 mật khẩu (Master Password) để mở “kho mật khẩu”
- Thẻ xác thực 2 yếu tố U2F Security Keys
  - Người dùng cần kết nối thẻ này với máy tính khi đăng nhập
  - Có khả năng giảm thiểu nguy cơ bị tấn công phishing
- Kích hoạt tùy chọn xác thực hai yếu tố (2FA) trên các hệ thống dịch vụ



## Tấn công vào hệ xác thực bằng mật khẩu

- Tấn công thụ động: nghe lén, quan sát quá trình nhập mật khẩu
  - Nhìn trộm
  - Sử dụng chương trình key logging
  - Chặn bắt gói tin trên đường truyền
  - Tấn công kênh bên
- Tấn công chủ động:
  - Đoán và thử
  - Giả mạo chương trình cung cấp dịch vụ (server)
  - Giả mạo chương trình khách (client)
  - Tấn công vào máy chủ cung cấp dịch vụ

## Tấn công đoán và thử (Online attack)

- Cách thức: dò thử lần lượt các mật khẩu, quan sát kết quả xác thực hệ thống trả lại
- Đặc điểm:
  - Tương tác trực tiếp với hệ xác thực
  - Có thể thử trên 1 hoặc đồng thời nhiều tài khoản
- Xác suất tấn công thành công:  $P \geq (T \times G)/N$ 
  - G: Tốc độ kẻ tấn công dò thử
  - T: Thời gian kẻ tấn công dò thử
  - N: Số mật khẩu hệ thống có thể tạo ra
- Giảm thiểu:
  - Tăng độ dài của mật khẩu
  - Quy định số lần thử xác thực tối đa trong một khoảng thời gian

## Tấn công lợi dụng lỗi hỏng lưu trữ (Offline attack)

- Lưu mật khẩu dưới dạng rõ:
    - Nguy cơ mất an toàn cao nhất
  - Lưu mật khẩu dưới dạng bản mã:
    - An toàn khi sử dụng hệ mật mã tốt, bảo vệ khóa giải mã an toàn
    - Hạn chế: nếu kẻ tấn công đánh cắp được khóa thì toàn bộ mật khẩu bị lộ
- Cần phải có một cách để xác thực mật khẩu mà không lưu trữ thông tin cho phép đối phương dễ dàng xác định được mật khẩu.

## Lưu bản mã

### Tạo tài khoản



User

password

Mã hóa

$k_s$ : được lưu trữ an toàn

Username

Encrypted password

### Xác thực



User

password

So sánh

$k_s$

Giải mã

Hệ thống



ĐẠI HỌC BÁCH KHOA HÀ NỘI  
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

13

13

## Băm mật khẩu

- Băm mật khẩu và lưu mã băm để xác thực:
  - Chi phí thấp hơn
  - Yêu cầu an toàn: hàm băm có tính một chiều
- Hạn chế:
  - Hàm băm có tính xác định: 2 mật khẩu giống nhau thì mã băm giống nhau  
→ Kẻ tấn công biết được 2 người dùng sử dụng mật khẩu giống nhau.
  - Tấn công từ điển/Tấn công cầu vồng: kẻ tấn công tạo bảng tra cứu (rainbow table) bằng cách băm tất cả mật khẩu mà hắn biết.



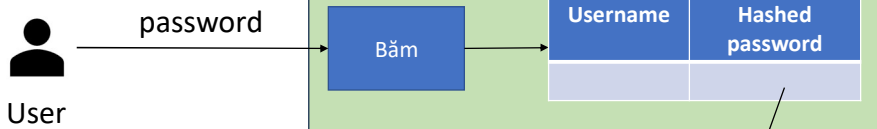
ĐẠI HỌC BÁCH KHOA HÀ NỘI  
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

14

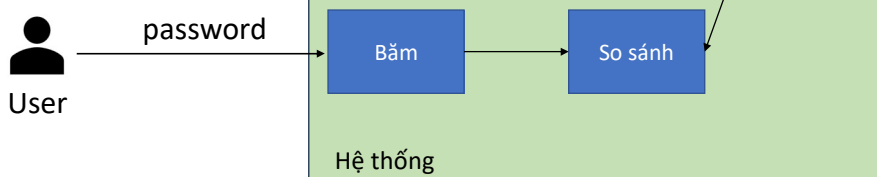
14

## Lưu mã băm mật khẩu

### Tạo tài khoản



### Xác thực



## Tấn công từ điển/Tấn công cầu vồng

### • Dictionary attack/Rainbow attack

Bảng tra cứu

Potential password	Hash(Potential password)
Abcdef	H1
123456	H2
Guessme	H3
Iloveyou	H4
Password	H5
Qwerty	H6

Bảng CSDL

Username	Hashed password
someone	H3
other	H1
...	...

Mật khẩu của tài khoản 'someone' là 'Guessme'

### • Giảm thiểu nguy cơ: Hash(Password, Salt)

- Yêu cầu sử dụng salt là gì?



giá trị ngẫu nhiên cho mỗi tài khoản

## Bấm mật khẩu với “salt” (1)

- Lưu trữ [salt, Hash(password || salt)]
- Ví dụ: Hệ điều hành Linux

username      salt      hash

bkcs: \$1\$J54g/weK\$aAVR2Nd6opP19kcUuTTgk.:17422:0:99999:7:::

algorithm  
 1: MD5-based  
 2: Blowfish  
 5: SHA-256  
 6: SHA-512

Lần cuối thay đổi (tính từ ngày 1/1/1970)  
 Số ngày tối thiểu trước khi đổi  
 Số ngày tối đa trước khi đổi  
 Số ngày trước khi hết hạn sẽ cảnh báo  
 Ngày hết hạn (tính từ 1/1/1970)

- Lưu trữ trong CSDL

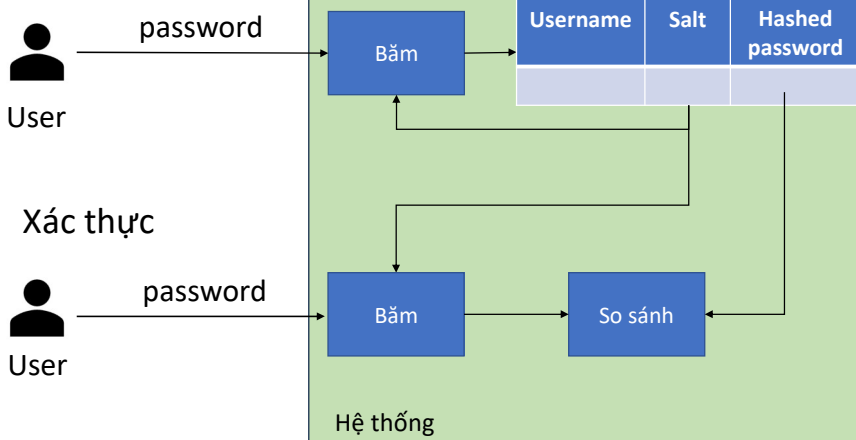
username	salt	hash
levn	iU9KjTeD	5myyo4W7zppTOEdVUeP8/E6Km...
tungbt	r.PhJ0HG	Y.xOpTBqJbWpc3f0uri.g8ErCu4wliUGq

17

salt -> khi cập nhật mới thì sẽ dùng hash

## Bấm mật khẩu với “salt” (2)

Tạo tài khoản



18

## Băm mật khẩu với “salt”: Tấn công cầu vồng

- Phải tính toán lại bảng tra cứu
- Mỗi bảng tra cứu ứng với 1 giá trị salt chỉ xác định được mật khẩu của tối đa 1 tài khoản
  - Dù hai người dùng sử dụng mật khẩu giống nhau nhưng giá trị băm vẫn khác nhau  
 $\text{Hash}(m, \text{salt}_1) \neq \text{Hash}(m, \text{salt}_2)$
  - Để tìm mật khẩu của tài khoản khác, kẻ tấn công phải tính toán lại bảng tra cứu.

19

threat ko gì m, impact gì m -> risk gì m

s d ng salt -> làm gì m áng k r i ro

risk = threat \* impact

## Băm mật khẩu với “salt”: Tấn công cầu vồng

Bảng CSDL

Username	Salt	Hashed password
someone	X	HX3
other	Y	
...	Z	

Bảng tra cứu

Potential password	Salt	Hash(Potential password, salt)
Abcdef	X	HX1
123456	X	HX2
Guessme	X	HX3
Iloveyou	X	HX4
Password	X	HX5
Qwerty	X	HX6
...	X	...

Mật khẩu của tài khoản 'someone' là 'Guessme'

- Mật khẩu dễ đoán thì dễ bị tìm ra hơn

20

## Băm mật khẩu với “salt” – Nâng cao an toàn

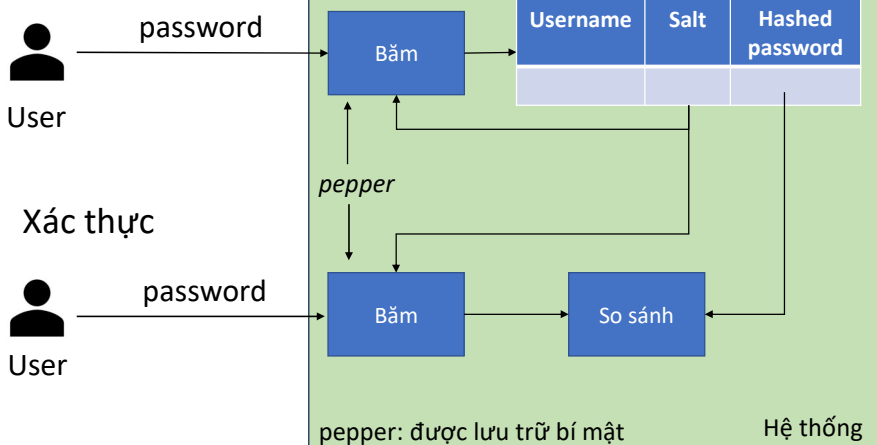
- Kẻ tấn công có thể tạo ra từ điển mới với các giá trị “salt”
- Băm nhiều lần:  $\text{hash}(\text{hash}(\dots\text{hash}(\text{password}, \text{salt}))))$ 
  - Mục đích: làm chậm thời gian tính toán giá trị xác thực  
→ làm chậm thời gian tấn công dò tìm...
  - ...nhưng kẻ tấn công có thể kiên nhẫn hơn nữa tạo ra từ điển mới
- Băm mật khẩu với một giá trị “pepper” bí mật
  - Mục đích: ngăn chặn kẻ tấn công tạo ra từ điển mới
- Đề xuất: Sử dụng một trong thuật toán bcrypt, scrypt, PBKDF2 thay cho các hàm băm thông thường

21      nâng cao an toàn h n-> thêm mã m t

pepper nên c l u tr trong các ph n c ng chuyên d ng

## Băm cùng salt + pepper

Tạo tài khoản



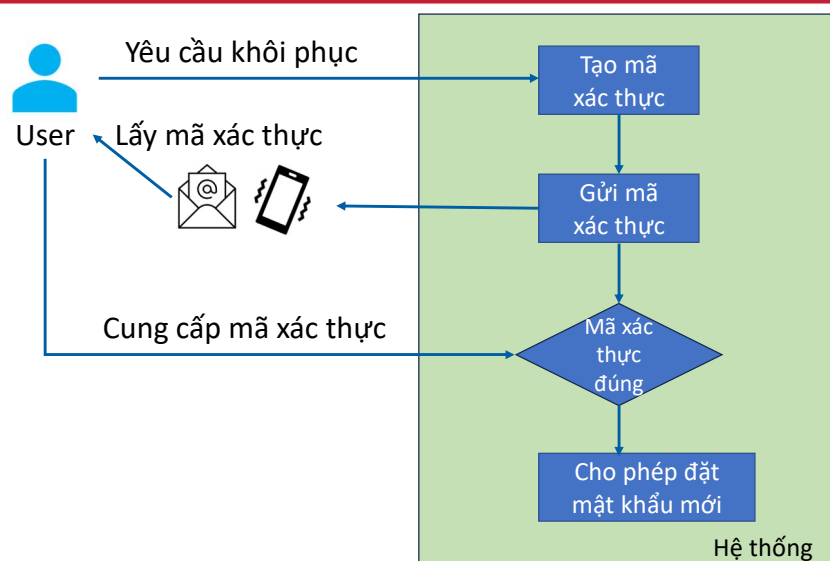
22

## Khôi phục mật khẩu

- Làm thế nào để người dùng có thể khôi phục mật khẩu khi họ quên?
  - Gửi trực tiếp qua email
  - Câu hỏi bí mật
  - Gửi mã khôi phục qua email
  - Gửi mã khôi phục qua tin nhắn SMS
  - ...
- Cơ chế khôi phục không được dễ dàng hơn cơ chế xác thực để kẻ tấn công vượt qua

23

## Khôi phục mật khẩu bằng mã xác thực



24

## Sử dụng câu hỏi bí mật còn an toàn?

- Năm 2008, ứng viên Phó Tổng thống Hoa Kỳ Sarah Palin bị đánh cắp tài khoản Yahoo Mail

The screenshot shows the Yahoo! account recovery process. It includes a progress bar with steps: 'What did you forget?', 'Verify your identity', and 'Reset your password'. The current step is 'Verify your identity', which asks the user to answer questions to validate their identity. The questions include: 'Birthday' (February 11), 'Country of Residence' (United States), 'Postal Code' (99654), and 'Where did you meet your spouse?' (Vassila High). There are 'Exit Wizard' and 'Next' buttons.

- Năm 2012, ứng viên Tổng thống Mitt Romney bị đánh cắp tài khoản Hotmail

## Một số chính sách sử dụng mật khẩu

- Mục đích: tăng cường an toàn cho hệ xác thực dựa trên mật khẩu
- Quy định độ dài tối thiểu
- Quy định các ký tự bắt buộc phải sử dụng
- Thay đổi mật khẩu định kỳ
- Hạn chế sử dụng lại mật khẩu cũ trong một khoảng thời gian nhất định
- Hạn chế số lần thử xác thực
- Tăng thời gian chờ thử xác thực lại
- Yêu cầu đổi mật khẩu sau lần đăng nhập đầu tiên
- Tuy nhiên, luôn phải cân nhắc sự trả giá cho tính tiện lợi

## 2. Một số giao thức xác thực

ONE LOVE. ONE FUTURE.

27

27

### Giao thức PAP

- Password Authentication Protocol
- Được sử dụng trong giao thức mạng PPP trước đây
- Nội dung:
  - (1)  $U \rightarrow S$ : ID || Password
  - (2) Server kiểm tra trong CSDL
  - $S \rightarrow U$ : ACK/NAK
- Không an toàn



ĐẠI HỌC BÁCH KHOA HÀ NỘI  
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

28

28

## Giao thức CHAP

- Challenge Handshake Authentication Protocol
- (1)  $U \rightarrow S$ : Request
  - (2)  $S \rightarrow U$ : Challenge
  - (3)  $U \rightarrow S$ :  $ID \parallel \text{Hash}(ID \parallel \text{Hash}(\text{Password}) \parallel \text{Challenge})$
  - (4) Server kiểm tra  
 $S \rightarrow U$ : ACK / NAK
- Challenge: chuỗi ký tự ngẫu nhiên
  - Hash: MD5

## Xác thực 1 chiều dựa trên hệ mật mã KCK

- (1)  $A \rightarrow B$ : Request *ISO/IEC 9798-3 / FIPS-196*
- (2)  $B \rightarrow A$ :  $\text{TokenID} \parallel N_B$
- (3)  $A \rightarrow B$ :  $\text{TokenID} \parallel \text{Cert}_A \parallel \text{TokenAB}$

TokenID: chứa thông tin của phiên

$$\text{TokenAB} = N_A \parallel N_B \parallel E(K_{RA}, N_A \parallel N_B)$$

Chữ ký số

## Giao thức EAP

- Extensible Authentication Protocol
- Có khoảng 40 biến thể kết hợp thêm nhiều cơ chế khác nhau:
  - EAP-MD5: tương tự CHAP
  - EAP-TLS, EAP-TTLS, PEAP: kết hợp TLS
  - EAP-POTP: kết hợp One-Time-Password
  - EAP-PSK: kết hợp pre-shared key
- ...

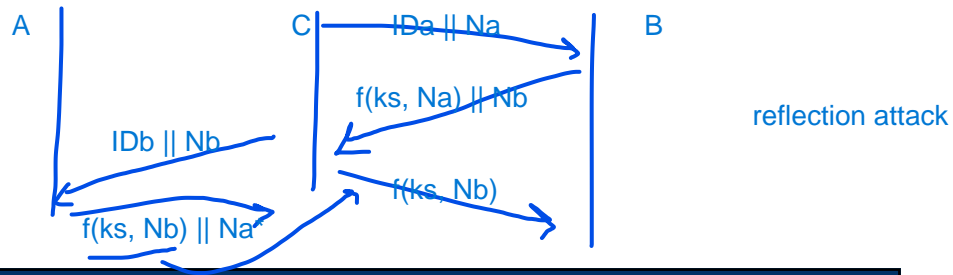
31

## Xác thực 2 chiều sử dụng hệ mật mã KĐX

- Giả sử A và B đã chia sẻ khóa  $K_S$
  - (1)  $A \rightarrow B: ID_A$
  - (2)  $B \rightarrow A: N_B$
  - (3)  $A \rightarrow B: f(K_S, N_B) || N_A$
  - (4)  $B \rightarrow A: f(K_S, N_A)$
- Hàm  $f$ : có thể là các hàm mã hóa KĐX, hàm băm  
 $K_S$ : khóa hoặc mật khẩu

32





## Bài tập

- Xem xét tính an toàn của giao thức xác thực sau:
  - (1)  $A \rightarrow B: ID_A || N_A$
  - (2)  $B \rightarrow A: f(K_S, N_A) || N_B$
  - (3)  $A \rightarrow B: f(K_S, N_B)$
- Nhận xét: người bắt đầu giao dịch phải là người chứng minh trước

## Xác thực 2 chiều sử dụng hệ mật mã KCK

- (1)  $A \rightarrow B: \text{Request}$  *ISO/IEC 9798-3 / FIPS-196*
- (2)  $B \rightarrow A: \text{TokenID} || N_B$
- (3)  $A \rightarrow B: \text{TokenID} || \text{Cert}_A || \text{TokenAB}$
- (4)  $B \rightarrow A: \text{TokenID} || \text{Cert}_B || \text{TokenBA}$

$$\text{TokenAB} = N_A || N_B || E(K_{RA}, N_A || N_B)$$

$$\text{TokenBA} = N_A || N_B || E(K_{RB}, N_A || N_B)$$

## Giao thức dạng zero-knowledge (ZKP)

- Peggy có bí quyết để phân biệt được các loại rượu vang khác nhau
- Victor muốn bỏ tiền để mua lại bí quyết
- Làm thế nào để Peggy chứng minh với Victor mà không làm lộ bí quyết?



## Giao thức ZKP (Đọc thêm)

- Là các giao thức cho phép một bên chứng minh được thông tin của mình mà không làm lộ nội dung thông tin đó cho các bên còn lại (bên thứ 2 hoặc kẻ tấn công)
- Các bên tham gia giao thức:
  - Peggy-Người chứng minh: Peggy nắm được một số thông tin nào đó và muốn chứng minh cho Victor nhưng không muốn để lộ thông tin này
  - Victor-Người thẩm tra: Được quyền hỏi một số câu hỏi đến khi chắc chắn Peggy nắm thông tin. Victor không thể đoán thông tin từ câu trả lời của Peggy, hoặc do cố tình lừa Peggy tiết lộ thông tin
  - Eve-Kẻ nghe lén: Giao thức cần chống lại việc Eve nghe lén thông tin
  - Mallory: có nhiều quyền hơn Eve, có thể nghe lén, sửa đổi bản tin hoặc phát lại bản tin

## Một ví dụ - Giao thức Feige-Fiat-Shamir

- Khởi tạo: Peggy chọn  $p, q$  là 2 số nguyên tố:
  - Tính  $n = p \times q$
  - Chọn  $s$  sao cho  $\text{UCLN}(s, n) = 1$ ,  $v$  sao cho  $v = s^2 \bmod n$
  - Công bố  $(n, v)$ . Peggy cần chứng minh cho Victor biết mình nắm giữ giá trị  $s$
- Giao thức:
  - (1)  $P \rightarrow V: x = r^2 \bmod n$        $r$ : số ngẫu nhiên
  - (2)  $V$  chọn ngẫu nhiên  $b \in \{0, 1\}$   
 $V \rightarrow P: b$
  - (3)  $P \rightarrow V: y = r \times s^b \bmod n$
  - (4)  $V$  kiểm tra phương trình đồng dư  $y^2 \equiv x \times v^b \pmod{n}$Hoặc viết dưới dạng khác  $y^2 \bmod n = x \times v^b \bmod n$

## Giả mạo

- Mallory có thể giả mạo bằng 2 cách:
  - (1) Bắt các cặp giá trị  $(x, y)$  và phát lại
  - (2) Phán đoán giá trị của bit  $b$  mà Victor thử thách:
    - Đoán  $b = 0$ , Mallory gửi  $x = r^2 \bmod n$  và  $y = r \bmod n$
    - Đoán  $b = 1$ , Mallory chọn  $y$  trước và tính  $x$  sao cho
$$y^2 \equiv x \times v \pmod{n}$$
- Xác suất thành công của Mallory là bao nhiêu?
- Làm thế nào để giảm xác suất thành công của Mallory trong 1 vòng kiểm tra?

## Nhận xét

- Vì Peggy nắm được giá trị của  $s$  nên có thể qua được vô số vòng kiểm tra (Tính đầy đủ - Completeness)
- Nếu Mallory không biết  $s$ , thì xác suất giả mạo thành công lớn nhất là  $2^{-n}$  với  $n$  là số vòng kiểm tra (Tính vững chãi- Soundness)
- Mallory không thể sử dụng lại bộ số  $(x,y)$  để lừa Victor
- Victor không biết gì về  $s$  vì bài toán tính căn bậc 2 rời rạc là khó
- Tương tự, Eve nghe trộm được mọi bộ số  $(x,y,b)$  cũng không thể đoán được  $s$



## Các nguy cơ

- Peggy không thay đổi  $r$  sau mỗi vòng kiểm tra
- Chess Grandmaster Problem
- Mafia Problem
- Terrorist Problem



### Giao thức ZKP dựa trên hệ mật mã RSA (Một ví dụ khác)

- Peggy có khóa công khai  $K_U = (e, n)$  cần chứng minh anh ta có bí mật  $m$
  - Khởi tạo: Peggy tính  $c = m^e \bmod n$
  - Giao thức:
    - (1)  $P \rightarrow V: x = r^e \bmod n$        $r$ : số ngẫu nhiên
    - (2)  $V$  chọn ngẫu nhiên  $b \in \{0, 1\}$   
 $V \rightarrow P: b$
    - (3)  $P \rightarrow V: y = r \times m^b \bmod n$
    - (4)  $V$  kiểm tra phương trình đồng dư  $y^e \equiv x \times c^b \pmod{n}$
- Tự kiểm tra tính đầy đủ và bền vững của giao thức.  
Hãy đọc thêm lý thuyết tổng quan về ZKP trong tài liệu.



ĐẠI HỌC BÁCH KHOA HÀ NỘI  
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

41

41

## 3. One Time Password (OTP)

ONE LOVE. ONE FUTURE.

42

42

## Xác thực đa yếu tố

- Phương pháp xác thực sử dụng mật khẩu không đủ an toàn
- Sử dụng mật khẩu một cách an toàn:
  - Đủ dài và khó đoán
  - Không dùng chung cho nhiều tài khoản
  - Thay đổi thường xuyên
  - ... → hầu hết người dùng không thực hiện được
- Kẻ tấn công có khả năng đánh cắp CSDL và thực hiện tấn công từ điển
- Xác thực đa yếu tố (MFA – Multi Factor Authentication)
  - Cái người dùng biết: mật khẩu
  - Yếu tố còn lại: tài sản, mẫu sinh trắc→ kẻ tấn công chỉ biết mật khẩu là không đủ để vượt qua xác thực



ĐẠI HỌC BÁCH KHOA HÀ NỘI  
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

43

43

## One Time Password

- Mật khẩu chỉ dùng để xác thực cho 1 phiên hoặc 1 giao dịch  
OTP: ng n -> thu n ti n chong i dùng
- Phân loại:
  - S/Key OTP sinh d a trên giá tr bí m t
  - Hash-based OTP (HOTP) sinh d a trên hàm b m
  - Time-based OTP (TOTP) sinh theo th i gian
- Cách thức phân phối:
  - SMS
  - Ứng dụng
  - Email
  - Token



ĐẠI HỌC BÁCH KHOA HÀ NỘI  
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

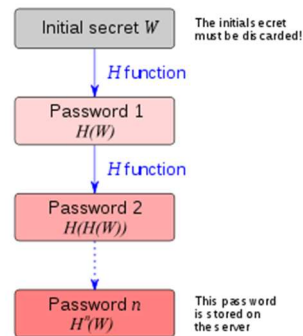
44

44

## S/Key OTP(RFC 1760)

- Sử dụng trong một số hệ điều hành Unix
- Pha sinh mật khẩu:
  - (1) Server chọn một giá trị bí mật  $W$
  - (2) Áp dụng hàm băm (hoặc HMAC) từ 1 đến  $n$  lần lên  $W$
  - (3) Lưu  $H_n$  trong CSDL
  - (4) Cung cấp cho client  $H_n, H_{n-1}, \dots, H_1$
  - (5) Client hủy giá trị  $H_n$

S/KEY password generation



## S/Key OTP(tiếp)

- Xác thực lần đầu
  - (1) Client gửi  $H_{n-1}$
  - (2) Server so sánh Hash( $H_{n-1}$ ) với  $H_n$  trong CSDL
  - (3) Nếu bước 3 xác thực đúng, thay  $H_n$  bằng  $H_{n-1}$ . Gửi thông báo xác thực thành công
  - (4) Client xóa  $H_{n-1}$  nếu đăng nhập thành công
- Xác thực các phiên kế tiếp: tương tự

## HOTP (RFC 4226)

- Bộ đếm:  $C$  (8 byte)
- Giá trị bí mật:  $K$  đã chia sẻ trước với client
- Hàm  $HOTP(K, C)$ 
  - (1) Tính  $HS = HMAC\text{-}SHA\text{-}1(K, C)$
  - (2) Trích xuất 4 bytes từ  $HS$  bằng hàm Dynamic Truncation
$$Sbits = DT(HS)$$
  - (3) Chuyển  $Sbits$  sang dạng thập phân. Lấy giá trị HOTP với số chữ số  $k$  tùy ý.
$$Snum = StToNum(Sbits)$$
$$D = Snum \bmod 10^k$$

## Hàm DT

- Đầu vào: Chuỗi 20 byte  $S$
- Xử lý:
  - Lấy  $OffsetBits = 4$  bit thấp của  $S[19]$
  - Biến đổi sang dạng thập phân  $Offset = StToNum(OffsetBits)$
  - Trích xuất 4 byte trong chuỗi  $S$  bắt đầu từ vị trí  $Offset$  được chuỗi  $P$
- Đầu ra: Xóa bit đầu tiên của  $P$



## Sử dụng HOTP trong giao thức xác thực

- Yêu cầu: Chia sẻ khóa K và C một cách an toàn
- Client:  $C \leftarrow C + 1$ . Tính HOTP(K, C) và người dùng gửi cho server
- Server:
  - Server:  $C \leftarrow C + 1$ . Tính HOTP(K, C) và kiểm tra
  - Nếu OTP nhận được là hợp lệ thông báo xác thực thành công
  - Nếu OTP nhận được không hợp lệ, thực hiện đồng bộ lại với tham số đồng bộ s. Yêu cầu xác thực lại.
  - Sau T lần xác thực lại không hợp lệ, khóa tài khoản

-> đồng bộ

## Đồng bộ trong HOTP

- Khi sử dụng HOTP trên thiết bị OTP Hardware Token, mã OTP được sinh ra theo yêu cầu người dùng
- Tình trạng mất đồng bộ: người dùng yêu cầu mã OTP nhưng không xác thực → giá trị bộ đếm của Token và Server khác nhau
- Đồng bộ hóa:
  - Server tính toán HOTP cho s lần kế tiếp
  - Yêu cầu người dùng gửi một chuỗi (2-3, hoặc hơn) các giá trị HOTP sinh được từ Token
  - So sánh chuỗi HOTP của người dùng với chuỗi HOTP đã sinh và thực hiện đồng bộ

số lần -> khi cần kết nối công an ứng  
OTP cao hơn

s d ng TOTP, ng i dùng ch có th s d ng 1 l n trong 1 phiên, nh ng phiên l n sau server ko áp ng  
c m i sau X giây -> b m C t ng 1 n v

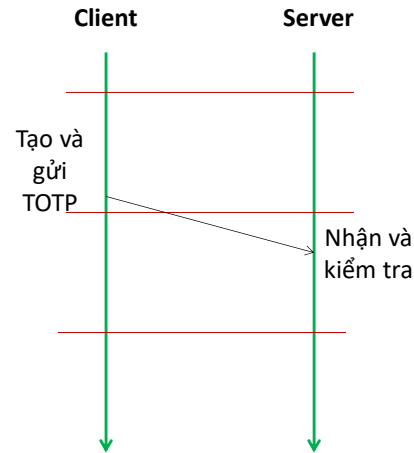
## TOTP(RFC 6238)

- Thực hiện tương tự HOTP
- Thay thế bộ đếm C bằng giá trị thời gian:  

$$C = (\text{Current UnixTime} - T_0)/X$$

$$T_0: \text{Mốc thời gian}$$

$$X: \text{Bước thời gian (time step)}$$
- Vấn đề trễ xử lý
- Client có thể gửi cùng 1 TOTP trong 1 bước thời gian, nhưng server chỉ chấp nhận cho 1 lần xác thực

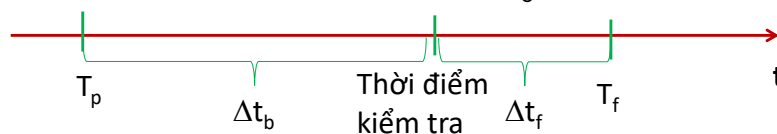


51

m t ng b th i gian: thu t toán ok -> chênh l ch d i 1s

## Mất đồng bộ trong TOTP

- Thời điểm tạo OTP ở client và kiểm tra ở server thuộc 2 chu kỳ khác nhau, hoặc
  - Mất đồng bộ đồng hồ
  - Phía kiểm tra cho phép chấp nhận một giá trị OTP nằm trong khoảng sai số cho phép
  - Miền chấp nhận  $[TOTP(T_p), TOTP(T_f)]$
- $$T_p = (\text{Current UnixTime} - 2X + 1 - T_0)/X$$
- $$T_f = (\text{Current UnixTime} + X - 1 - T_0)/X$$



52

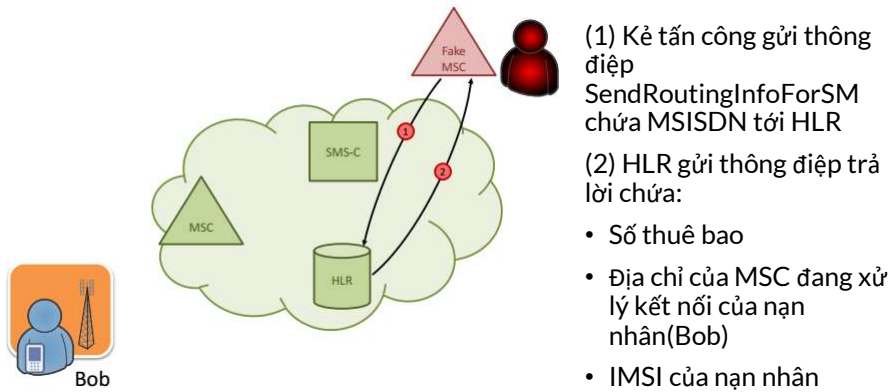
## SMS OTP

- Giá trị OTP được sinh ở server và gửi cho người dùng qua tin nhắn SMS
- Không đảm bảo an toàn:
  - Điện thoại người dùng bị nghe lén
  - Giả mạo trạm BTS
  - Tấn công lợi dụng lỗ hổng của giao thức SS7

## Tấn công lỗ hổng của SS7 (Đọc thêm)

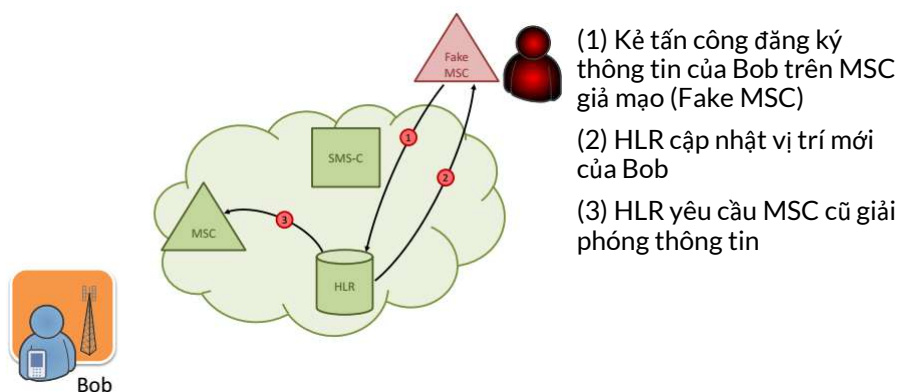
- SS7(Signaling System 7): bộ giao thức điều khiển truyền dữ liệu giữa các cell trong mạng di động
- Không có cơ chế xác thực
- IMSI: Định danh của thẻ SIM
- IMEI: Định danh của thiết bị
- MSISDN: Số thuê bao
- HLR(Home Location Register): CSDL thuê bao
- MSC(Mobile Switching Center): Bộ chuyển mạch
- MAP(Mobile Application Part): giao thức điều phối truyền dữ liệu giữa các thành phần trong phiên dịch vụ

## Tấn công SS7 – Bước 1



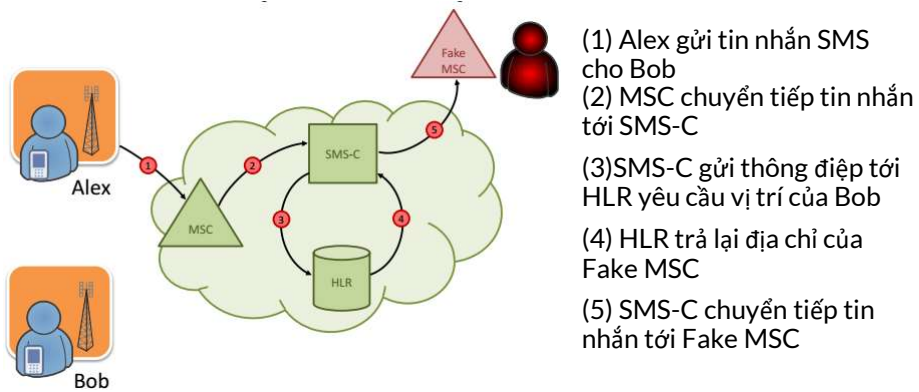
55

## Tấn công SS7 – Bước 2



56

## Tấn công SS7 – Bước 3



## Một vụ việc tấn công xác thực người dùng



**Vì sao khách hàng Vietcombank mất 500 triệu trong tài khoản?**

## Một vụ việc tấn công xác thực người dùng

Kịch bản sử dụng dịch vụ:

- B1: Khách hàng đăng nhập vào hệ thống eBanking
- B2: Khách hàng nhập lệnh chuyển tiền
- B3: Hệ thống eBanking gửi mã OTP qua tin nhắn SMS tới số điện thoại mà khách hàng đã đăng ký
- B4: Khách hàng nhập mã OTP nhận được vào hệ thống để xác nhận chuyển tiền

→Xác thực đa yếu tố:

- (1) Mật khẩu truyền thống
- (2) SMS OTP

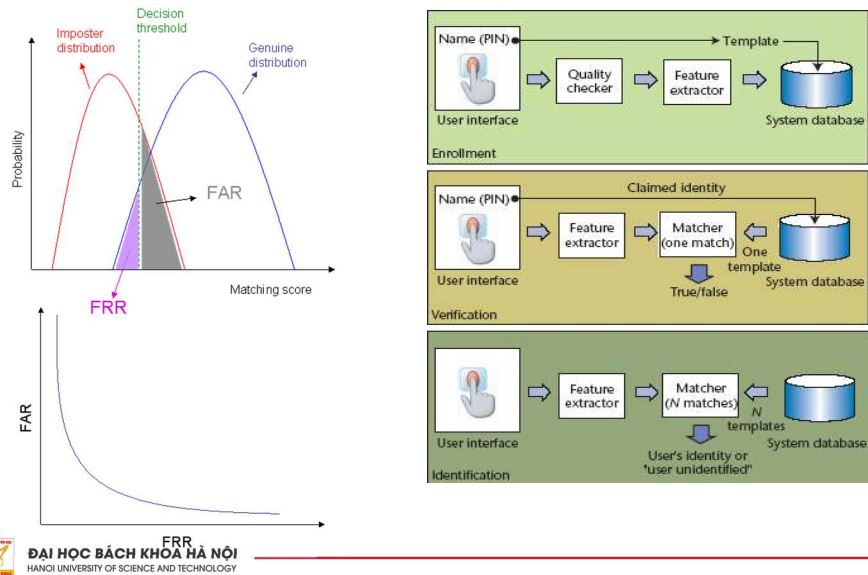
phishing attack: tấn công giả mạo

## Một vụ việc tấn công xác thực người dùng

- Vietcombank cung cấp ứng dụng di động Vietcombank Smart OTP cung cấp mã xác thực OTP
- B1: Mở ứng dụng và điền số ĐT đăng ký SMS Banking
- B2: Hệ thống gửi mã xác thực OTP tới số điện thoại
- B3: Người dùng nhập mã xác thực vào ứng dụng
- B4: Nếu mã OTP đúng, ứng dụng được kích hoạt



## Xác thực bằng sinh trắc (biometric)



61

61

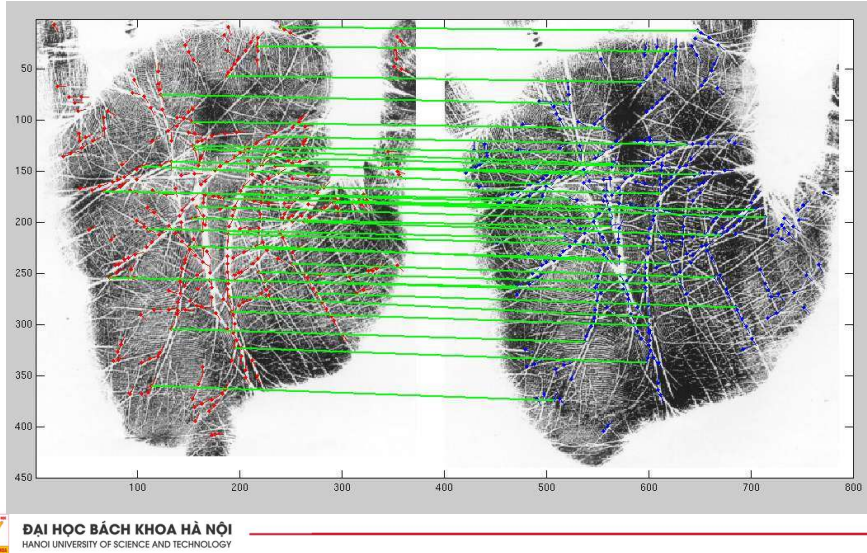
## Dấu vân tay



62

62

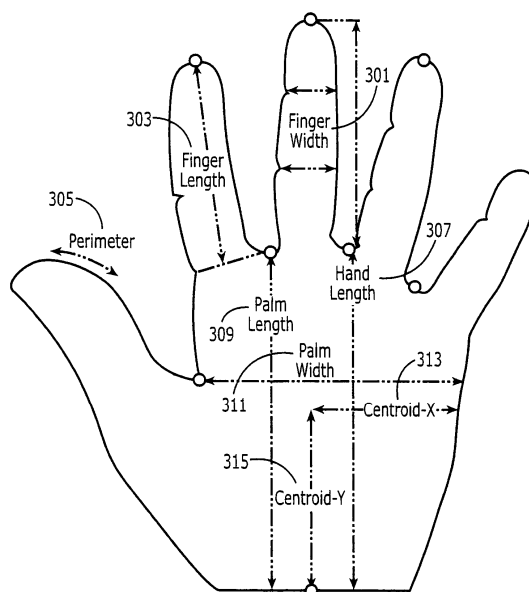
## Vân lòng bàn tay



63

63

## Cấu trúc bàn tay

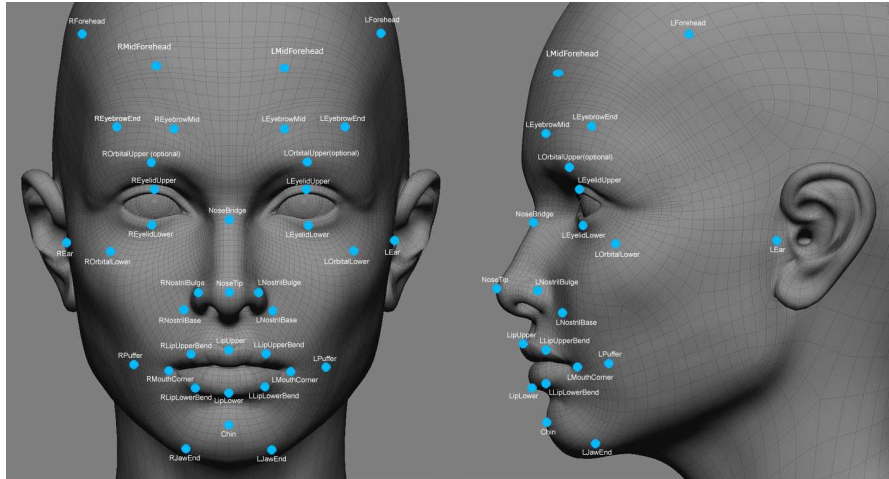


64

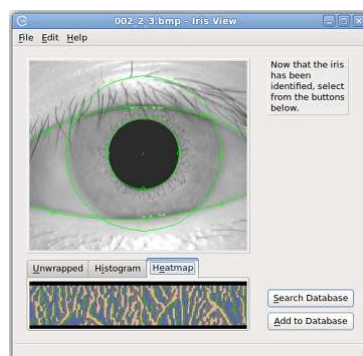
64



## Khuôn mặt



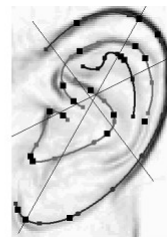
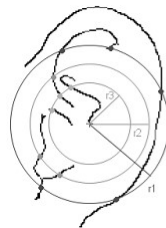
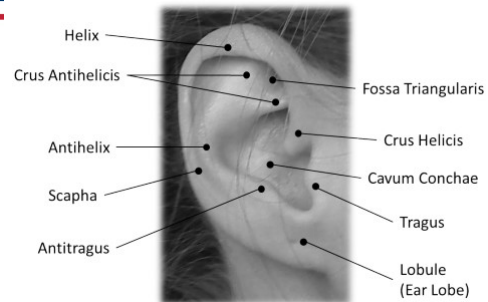
## Mống mắt



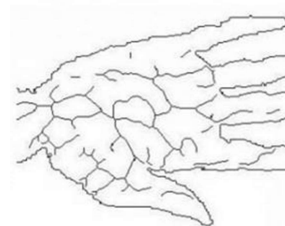
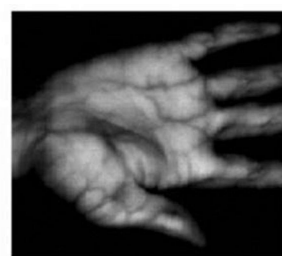
### HOW IRIS SCANNERS RECORD IDENTITIES

- 1 Scanner reads from outer iris inwards to pupil edge
- 2 Scanner plots distinct markings on iris and maps unique shape
- 3 After plotting many marks within the iris all data is saved to a database
- 4 Other scanners will compare this data to verify individual identities

## Vành tai



## Mạch máu



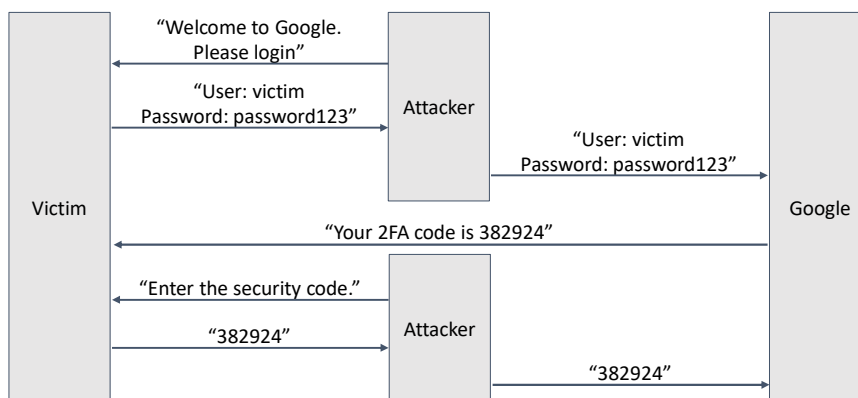
## Vượt qua xác thực đa yếu tố

- Tấn công chuyển tiếp(relay attack): sử dụng kỹ thuật giả mạo để đánh cắp các yếu tố
- Ví dụ:
  - Sơ đồ xác thực 2 yếu tố:
    - Yếu tố thứ 1: mật khẩu
    - Yếu tố thứ 2: mã OTP gửi tới điện thoại
  - Tấn công:
    - Website giả mạo yêu cầu người dùng nhập mật khẩu
    - Kẻ tấn công đăng nhập ngay vào website thật với mật khẩu đã có
    - Website thật gửi mã OTP tới điện thoại người dùng
    - Website giả mạo yêu cầu người dùng nhập mã OTP
    - Kẻ tấn công sử dụng mã OTP để hoàn thành xác thực bước 2

Passkey authentication

WebAuthentication -> passwordless authentication

## Tấn công chuyển tiếp – Ví dụ



## 4. Single Sign On(SSO)

ONE LOVE. ONE FUTURE.

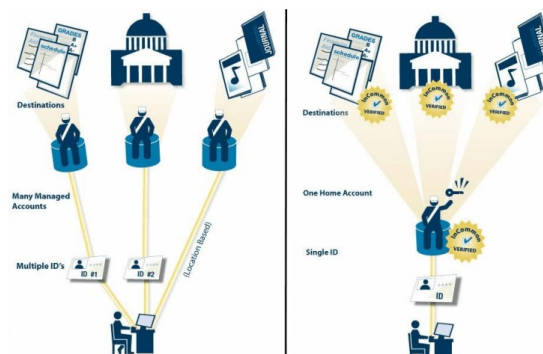
71

71

single account -> dùng 1 tài khoản cho nhiều ứng dụng

### Khái niệm

- SSO là một cơ chế xác thực yêu cầu người dùng đăng nhập vào chỉ một lần với một tài khoản và mật khẩu để truy cập vào nhiều ứng dụng trong 1 phiên làm việc (session).



ĐẠI HỌC BÁCH KHOA  
HANOI UNIVERSITY OF SCIENCE

KHÔNG CÓ SINGLE SIGN ON

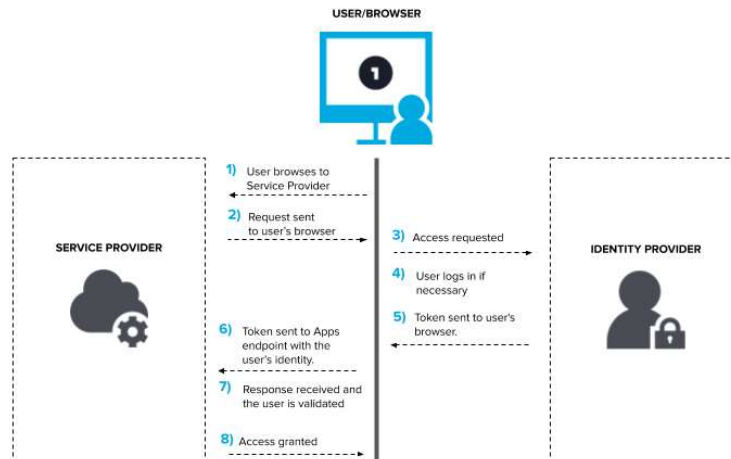
CÓ SINGLE SIGN ON

72

72

## Single Sign On

### Service Provider Initiated Workflow



## Các giải pháp SSO

- OpenID Connect
- CAS
- Open SAML
- CA Single Sign On
- Java Open Single Sign On
- Google Sign-In
- Facebook Login