Two-Factor Authentication

using TOTP

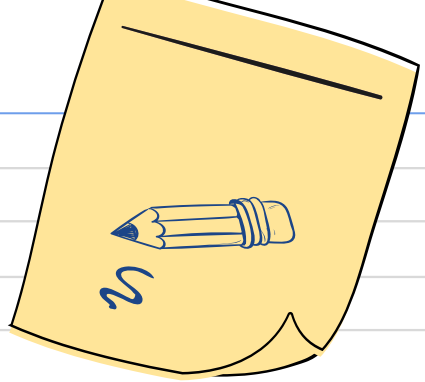
GVHD: Nguyễn Ngọc Tự

Bùi Hoàng Trúc Anh - 21521817
Nguyễn Ngọc Trà My - 21520353
Lê Hoàng Oanh – 21521253

Nội dung

1. Ngữ cảnh ứng dụng
2. Tổng quan về TOTP
3. Thuật toán
4. Quá trình tạo mã TOTP
5. Kết quả nghiên cứu
6. Chương trình demo

1. Ngữ cảnh ứng dụng

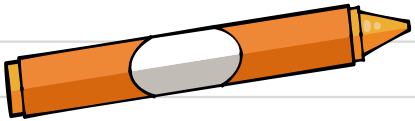


Để tăng tính bảo mật trong các ứng dụng chuyển tiền online, server sẽ yêu cầu người dùng nhập lại mật khẩu và tạo thêm mã TOTP để xác thực người dùng.

Những bên liên quan

Server

Các dịch vụ này yêu cầu người dùng sử dụng TOTP để bảo vệ tài khoản của họ.



User

Là người sử dụng TOTP để bảo vệ tài khoản trực tuyến của mình

Attacker

Giả sử attacker biết username và pass

Mục tiêu bảo mật

$a^2+b^2=c^2$

Đảm bảo tính xác thực

Đảm bảo rằng mã xác thực được tạo ra chỉ có thể được sử dụng một lần và chỉ có thể được tạo ra bởi người dùng cụ thể đã được cấp quyền truy cập

Đảm bảo tính bảo mật:

Đảm bảo rằng mã xác thực được tạo ra không thể bị đoán trước hoặc giả mạo bởi các kẻ xấu.

2. Tổng quan về TOTP

OTP

- (One-Time Password) là mật khẩu sử dụng một lần.

- Người dùng sẽ được cung cấp một mã OTP duy nhất để xác thực việc truy cập vào một tài khoản hoặc thực hiện một giao dịch cụ thể.

- Chỉ có thể sử dụng được một lần duy nhất và sẽ hết hạn sau một khoảng thời gian ngắn.

2. Tổng quan về TOTP

HOTP

(HMAC-based one-time password) là thuật toán mật khẩu dùng một lần dựa trên cơ chế xác thực thông điệp bằng hàm băm HMAC (Hash-based Message Authentication Code) và hàm băm SHA-1 (Secure Hash Algorithm 1).

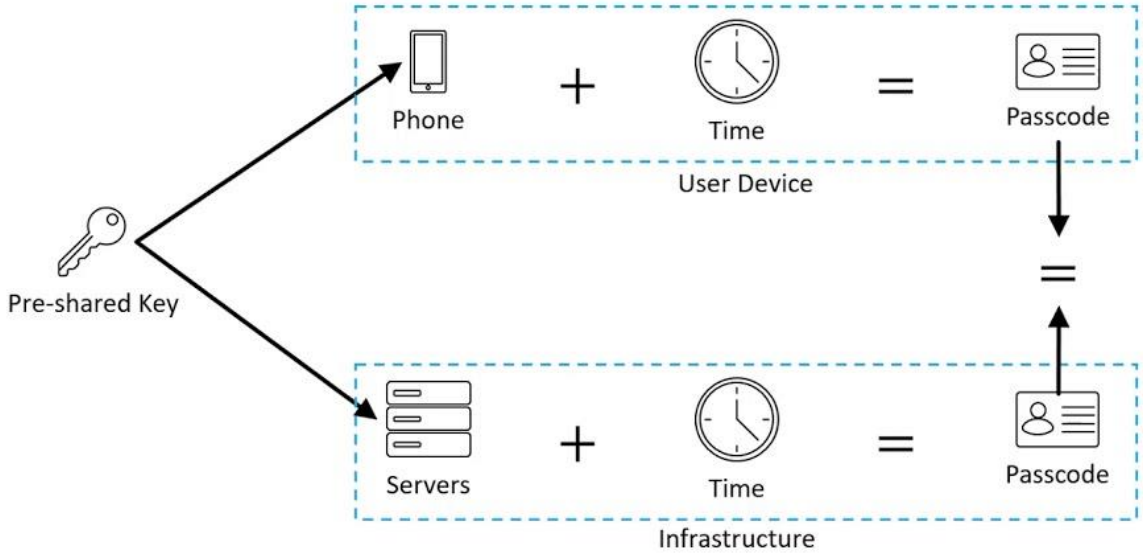
2. Tổng quan về TOTP

TOTP

là một thuật toán sử dụng để tạo ra mã xác thực một lần (OTP) dựa trên thời gian hiện tại và một khóa bí mật chia sẻ giữa người dùng và hệ thống. TOTP là một phiên bản mở rộng của HOTP (HMAC-based One-time Password).

Time-based One-Time Password

Overview



3. Thuật toán

Thay thế bộ đếm C của HOTP bằng thời gian thực T: TOTP = HOTP(K, T)

$$T = Floor\left(\frac{T_{current-unix-time} - T_0}{X}\right)$$

Trong đó:

Tcurrent-unix-time: giá trị thời gian hiện tại và được tính theo thời gian Unix (được tính từ thời điểm của Unix Epoch là ngày 01/01/1970 theo UTC - giờ chuẩn quốc tế).

T_0: giá trị thời gian ban đầu (thông thường sẽ chọn giá trị 0).

X: bước thời gian, tham số này được coi là yếu tố quyết định thời gian hợp lệ của mật khẩu OTP.

T: kết quả phép tính (lấy phần nguyên) và là giá trị cần tìm.

Độ dài mật khẩu của thuật toán TOTP:

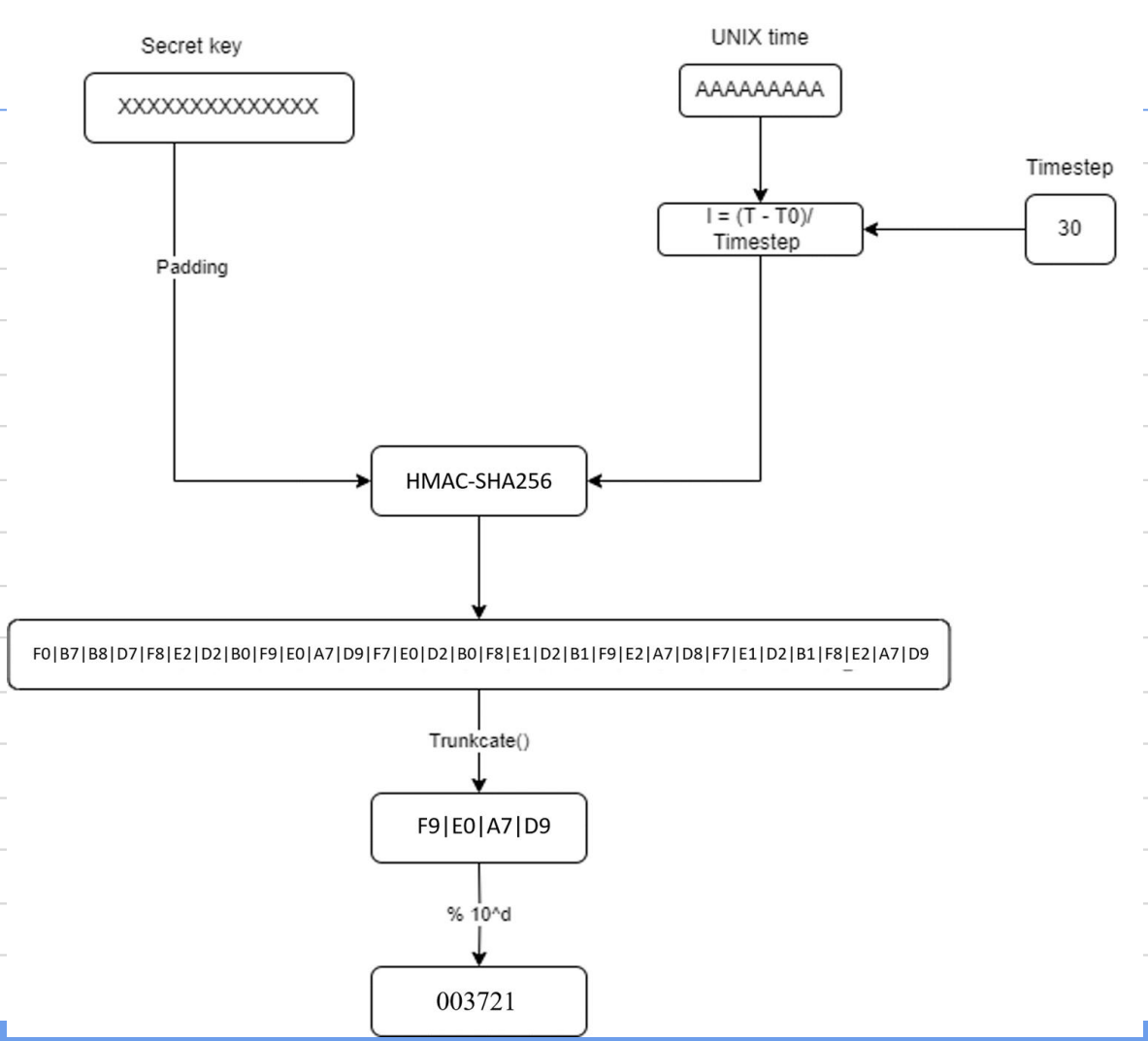
$$TOTPvalue = TOTP(K, T) \bmod 10^N$$

Trong đó:

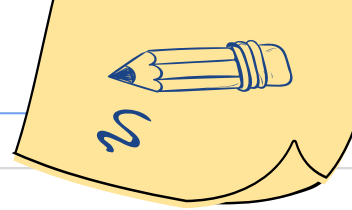
N: là số chữ số của OTP, thông thường một mật khẩu OTP sinh ra có độ dài từ 6 đến 8 chữ số (ví dụ là: 123456,…).

Với thuật toán TOTP thì thời gian chuẩn hợp lệ của mật khẩu OTP là 30 giây (X=30), thời gian có hiệu lực của mỗi lần sử dụng mật khẩu OTP là 30 giây được chọn phù hợp với yêu cầu về bảo mật và khả năng sử dụng.

4. Quá trình tạo mã TOTP

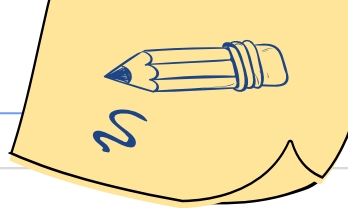


5. Kết quả nghiên cứu



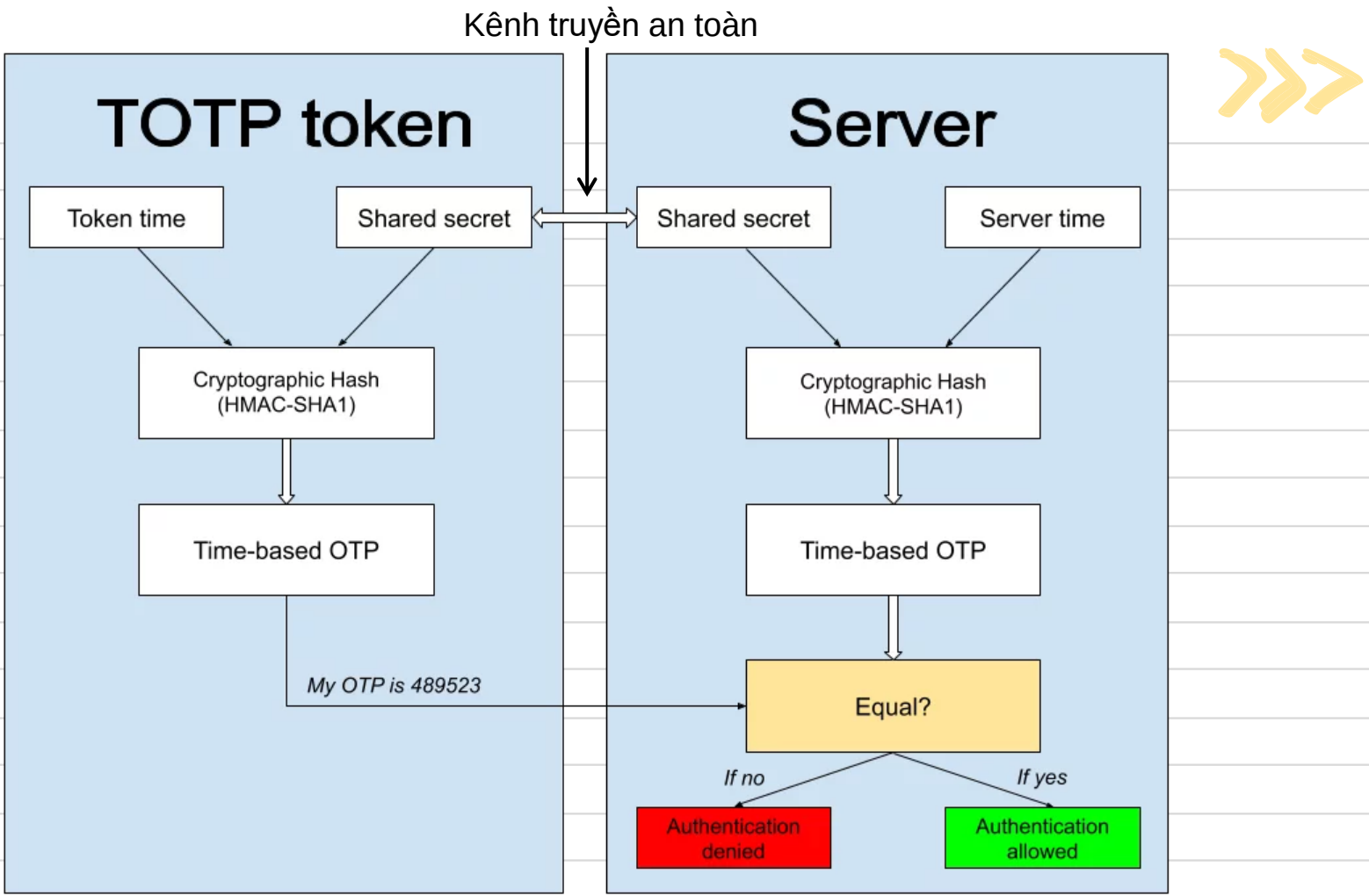
- Tìm hiểu thuật toán được sử dụng trong TOTP
- Xây dựng được chương trình mô phỏng lại thuật toán, sinh ra mã OTP để tăng tính xác thực

6. Chương trình demo

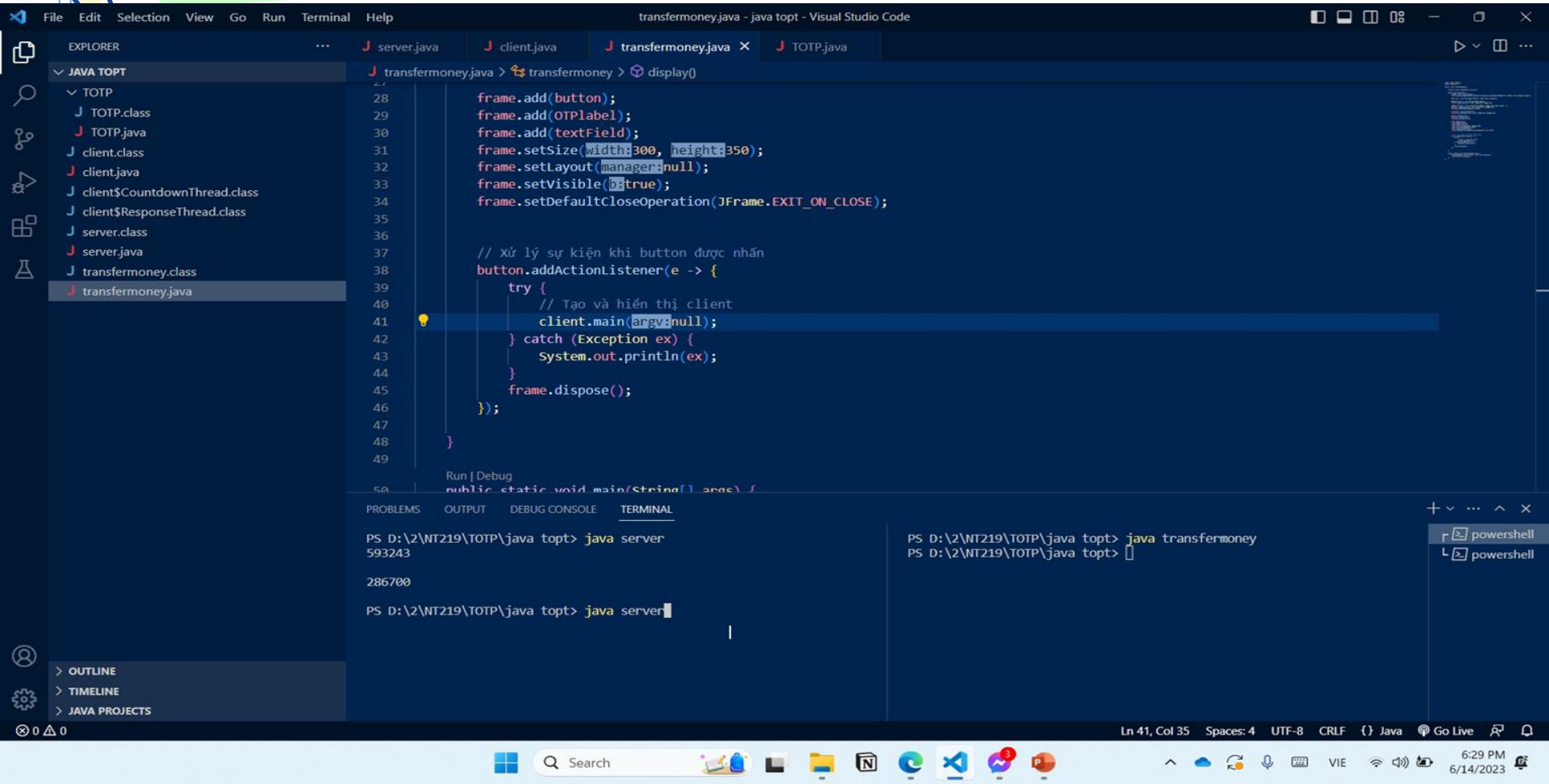


Kịch bản ứng dụng

Xây dựng 1 local server và 1 ứng dụng cho client trên window. Khi client nhập mật khẩu để xác thực yêu cầu chuyển tiền, ứng dụng sẽ tự động sinh ra một mã OTP. Sau đó ứng dụng sẽ gửi mã OTP này cùng với mật khẩu đến server để xác thực client.


Kênh truyền an toàn
TOTP token
Server
Token time
Shared secret
Shared secret
Server time
Cryptographic Hash (HMAC-SHA1)
Cryptographic Hash (HMAC-SHA1)
Time-based OTP
Time-based OTP
My OTP is 489523
Equal?
If no
If yes
Authentication denied
Authentication allowed

Demo

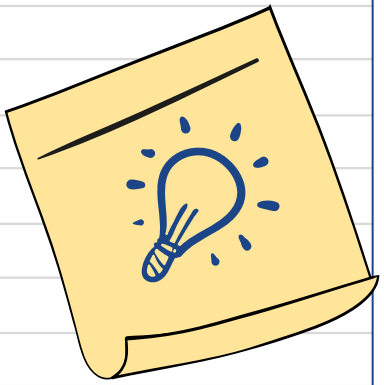

File Edit Selection View Go Run Terminal Help
transfermoney.java - java topt - Visual Studio Code
EXPLORER
JAVA TOPT
TOTP
TOTP.class
TOTP.java
client.class
client.java
client$CountdownThread.class
client$ResponseThread.class
server.class
server.java
transfermoney.class
transfermoney.java
server.java
client.java
transfermoney.java
TOTP.java
transfermoney.java > transfermoney > display()
frame.add(button);
frame.add(OTPlabel);
frame.add(textField);
frame.setSize(width:300, height:350);
frame.setLayout(manager:null);
frame.setVisible(b:true);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
// Xử lý sự kiện khi button được nhấn
button.addActionListener(e -> {
try {
// Tạo và hiển thị client
client.main(argv:null);
} catch (Exception ex) {
System.out.println(ex);
}
frame.dispose();
});
}
Run | Debug
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS D:\2\NT219\TOTP\java topt> java server
593243
286700
PS D:\2\NT219\TOTP\java topt> java server
PS D:\2\NT219\TOTP\java topt> java transfermoney
PS D:\2\NT219\TOTP\java topt>
powershell
powershell
OUTLINE
TIMELINE
JAVA PROJECTS
Ln 41, Col 35 Spaces: 4 UTF-8 CRLF {} Java Go Live
Search
VIE
6:29 PM
6/14/2023

Tài liệu tham khảo

- Balasta, D. U., Pelito, S. M. C., Blanco, M. C. R., Alipio, A. J., Mata, K. E., & Cortez, D. M. A. Enhancement of Time-Based One-Time Password for 2-Factor Authentication.

- Seta, H., Wati, T., & Kusuma, I. C. (2019, October). Implement time based one time password and secure hash algorithm 1 for security of website login authentication. In 2019 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS) (pp. 115-120). IEEE

- Dobreva, J., Lumburovska, L., Trpcheska, H. M., & Dimitrova, V. (2021). A Comparative Analysis of HOTP and TOTP Authentication Algorithms. Which one to choose?. Security & Future



Phân công

Nguyễn Ngọc Trà My	Lê Hoàng Oanh	Bùi Hoàng Trúc Anh
• Thuyết trình • Tìm hiểu thuật toán • Demo	• Ghi báo cáo đồ án • Tìm hiểu thuật toán	• Ghi báo cáo đồ án • Tìm hiểu thuật toán • Demo

End.