Somaiya Vidyavihar University
K J Somaiya School of Engineering

Course Name:	**Information Security (116U01L602)**	**Semester:**	VI
Date of Performance:	17 / 02 / 2025	**DIV/ Batch No:**	A-4
Student Name:	Hyder Presswala	**Roll No:**	16010122151

Title: Implementation of CAPTCHA for Security of systems

Objectives:

To write a program to convert plain text into cipher text using Caesar cipher and Transposition cipher

Expected Outcome of Experiment:

CO4 :- Illustrate and Compare network security mechanisms

Books/ Journals/ Websites referred:

1. Security in Computing
2. Cryptography and Network Security
3. Cryptography and Network Security: Principles and Practice

Pre Lab/ Prior Concepts:

New Concepts to be learned:

Abstract:

Implementation Details:

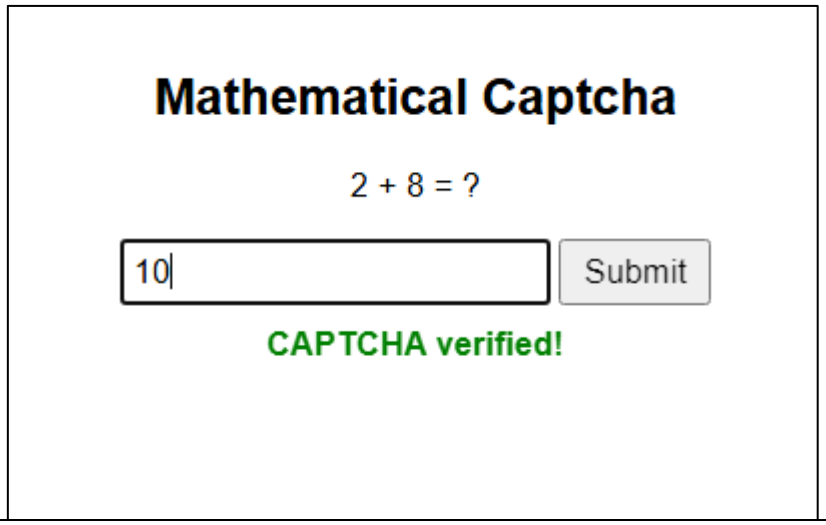
Three types of Captcha :-

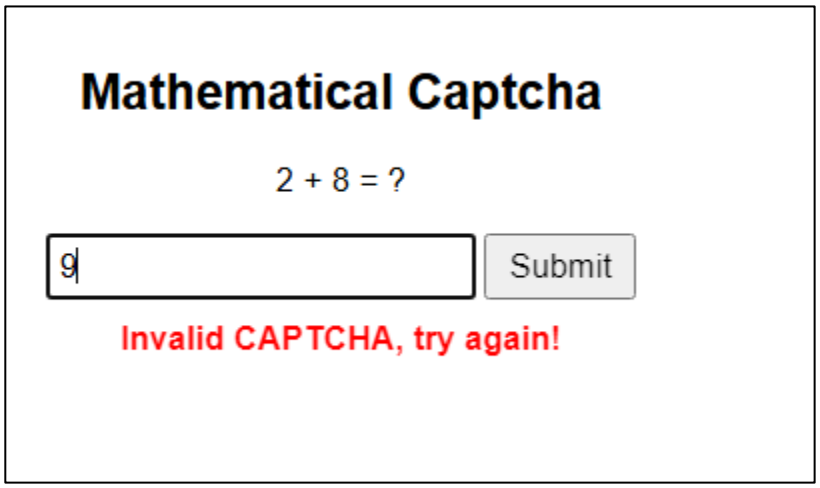
1. Mathematical Captcha
2. Images Captcha
3. Text Captcha

1) Mathematical Captcha:-

Link for Code :-
https://drive.google.com/drive/u/0/folders/1vgUUYbokp_E5ittzEIBoa3r_1d94tRnW

```
hyder@HyderPresswala MINGW64 ~/Downloads/Mathmatical
$ python app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 251-305-050
```



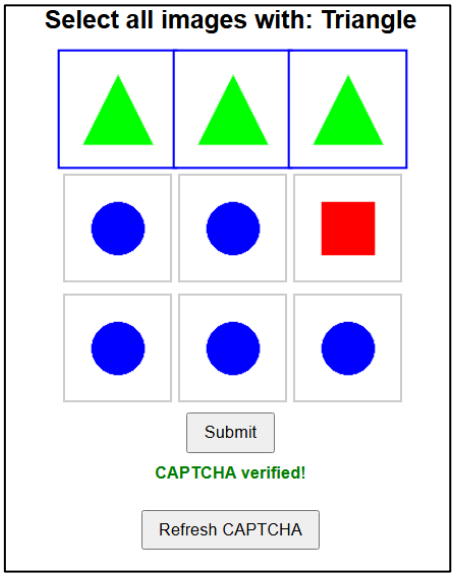


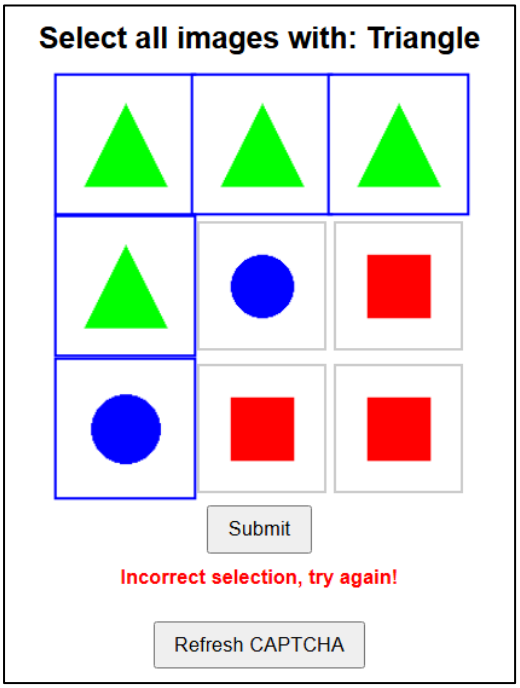
2) Images Captcha

Link for Code :-
https://drive.google.com/drive/u/0/folders/1vgUUYbokp_E5ittzEIBoa3r_1d94tRnW

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

hyder@HyderPresswala MINGW64 ~/Downloads/IS/Lab/Exp 03/The links may not work so here are the files/Images Captcha
$ python app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 251-305-050
127.0.0.1 - - [17/Feb/2025 23:33:10] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [17/Feb/2025 23:33:10] "GET /static/script.js HTTP/1.1" 200 -
127.0.0.1 - - [17/Feb/2025 23:33:10] "GET /static/styles.css HTTP/1.1" 200 -
127.0.0.1 - - [17/Feb/2025 23:33:11] "GET /generate_captcha HTTP/1.1" 200 -
127.0.0.1 - - [17/Feb/2025 23:33:11] "GET /captcha_images/captcha_1.png?t=1739815391063 HTTP/1.1" 200 -
127.0.0.1 - - [17/Feb/2025 23:33:11] "GET /captcha_images/captcha_0.png?t=1739815391063 HTTP/1.1" 200 -
127.0.0.1 - - [17/Feb/2025 23:33:11] "GET /captcha_images/captcha_2.png?t=1739815391063 HTTP/1.1" 200 -
127.0.0.1 - - [17/Feb/2025 23:33:11] "GET /captcha_images/captcha_3.png?t=1739815391063 HTTP/1.1" 200 -
127.0.0.1 - - [17/Feb/2025 23:33:11] "GET /captcha_images/captcha_4.png?t=1739815391063 HTTP/1.1" 200 -
127.0.0.1 - - [17/Feb/2025 23:33:11] "GET /captcha_images/captcha_5.png?t=1739815391063 HTTP/1.1" 200 -
127.0.0.1 - - [17/Feb/2025 23:33:11] "GET /captcha_images/captcha_6.png?t=1739815391063 HTTP/1.1" 200 -
127.0.0.1 - - [17/Feb/2025 23:33:11] "GET /captcha_images/captcha_7.png?t=1739815391063 HTTP/1.1" 200 -
127.0.0.1 - - [17/Feb/2025 23:33:11] "GET /captcha_images/captcha_8.png?t=1739815391063 HTTP/1.1" 200 -
127.0.0.1 - - [17/Feb/2025 23:33:22] "POST /verify_captcha HTTP/1.1" 200 -
127.0.0.1 - - [17/Feb/2025 23:33:22] "GET /generate_captcha HTTP/1.1" 200 -
```

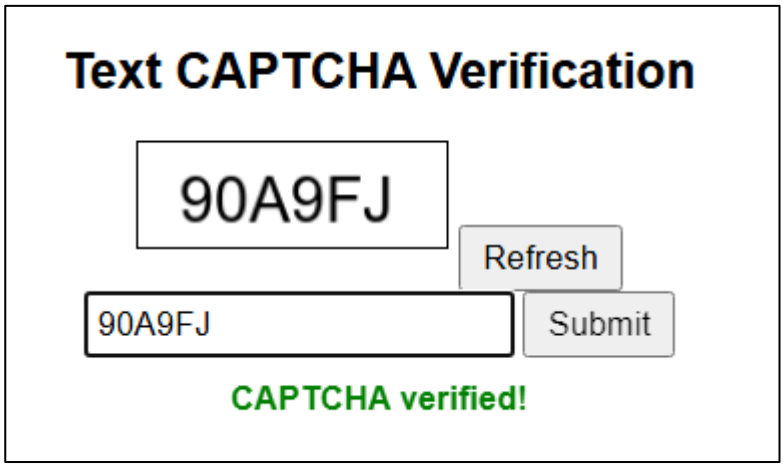


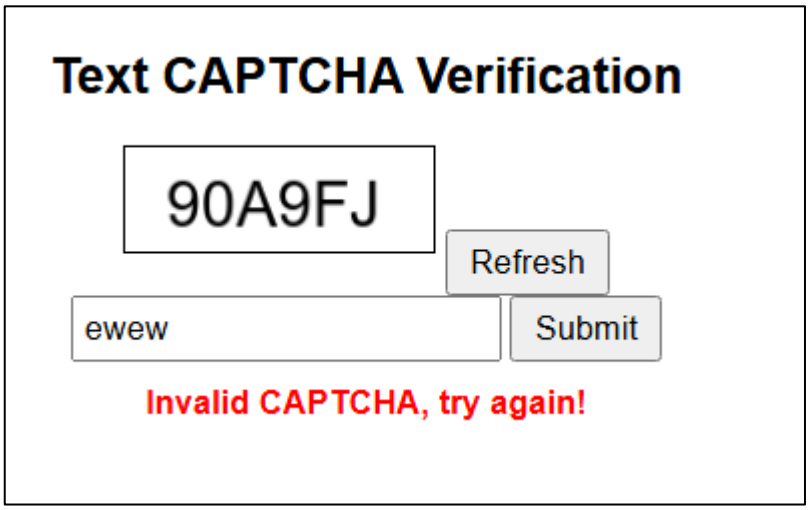


3) Text Captcha

Link for Code :-
https://drive.google.com/drive/u/0/folders/1vgUUYbokp_E5ittzEIBoa3r_1d94tRnW

```
hyder@HyderPresswala MINGW64 ~/Downloads/IS/Lab/Exp 03/The links may not work so here are the files/Images Captcha
$ python app.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with watchdog (windowsapi)
 * Debugger is active!
 * Debugger PIN: 251-305-050
127.0.0.1 - - [17/Feb/2025 22:55:25] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [17/Feb/2025 22:55:25] "GET /static/styles.css HTTP/1.1" 200 -
127.0.0.1 - - [17/Feb/2025 22:55:25] "GET /static/script.js HTTP/1.1" 200 -
127.0.0.1 - - [17/Feb/2025 22:55:26] "GET /generate_captcha HTTP/1.1" 200 -
127.0.0.1 - - [17/Feb/2025 22:55:26] "GET /captcha_image?1739813126163 HTTP/1.1" 200 -
127.0.0.1 - - [17/Feb/2025 22:55:31] "GET /generate_captcha HTTP/1.1" 200 -
127.0.0.1 - - [17/Feb/2025 22:55:31] "GET /captcha_image?1739813131351 HTTP/1.1" 200 -
```





Conclusion:

Thus, from this experiment we learnt about different types of Captcha's, their limitations, applications and also implemented them.

Post-Lab Questions:

1. Discuss how CAPTCHA helps in improving web security.

   CAPTCHA (**Completely Automated Public Turing test to tell Computers and Humans Apart**) helps websites stay secure by distinguishing between real users and bots.
   - **Prevents Brute Force Attacks** – Stops bots from guessing passwords.
   - **Blocks Spam & Fake Registrations** – Prevents fake accounts and comment spam.
   - **Stops Web Scraping** – Protects sensitive data from being stolen.
   - **Defends Against DDoS Attacks** – Reduces bot traffic to keep websites running smoothly.
   - **Secures Online Polls & Forms** – Ensures only real users can vote or submit data.
   - **Prevents Fraud & Fake Reviews** – Stops bots from posting fake reviews or making fraudulent transactions.

2. What are the different types of CAPTCHA? Is re-CAPTCHA different from CAPTCHA? Justify your answer.

Types of CAPTCHA

- **Text-Based CAPTCHA** – Users type distorted text from an image.
- **Math CAPTCHA** – Solving a simple math problem (e.g., 5 + 3 = ?).
  Image-Based CAPTCHA – Selecting specific objects from images (e.g., "Click on all traffic lights").
- **Audio CAPTCHA** – Listening to a distorted sound and typing the spoken words.
- **reCAPTCHA (Google's CAPTCHA)** – Uses AI to verify humans with minimal effort, often just a checkbox ("I'm not a robot").
- **Invisible CAPTCHA** – Works in the background, detecting bots based on user behavior without requiring input.

Yes, reCAPTCHA is an advanced form of CAPTCHA.

- **CAPTCHA** is a general term for any test that differentiates humans from bots.
- **reCAPTCHA** (by Google) is a more advanced version that uses AI and machine learning to improve security **while reducing user effort**.

For example, **reCAPTCHA v2** asks users to click a checkbox, and **reCAPTCHA v3** works silently in the background to detect bots based on behavior.

3. List the limitations of CAPTCHA. Mention the alternatives to CAPTCHA which can overcome these limitations.

Limitations of CAPTCHA

- **User Frustration** – Complex CAPTCHAs can be difficult to solve, leading to poor experience.
- **Accessibility Issues** – Visually impaired users struggle with text/image CAPTCHAs; audio CAPTCHAs can be unclear.
- **AI & Bots Bypass** – Advanced AI can now solve traditional CAPTCHAs with high accuracy.
- **Time-Consuming** – Users may find solving CAPTCHAs annoying, slowing down interactions.
- **Mobile Unfriendliness** – Small screens make CAPTCHAs harder to solve, reducing usability.
- **Language Barriers** – Some text-based CAPTCHAs contain unfamiliar words.

Alternatives to CAPTCHA

- **Honeypots** – Hidden fields in forms that only bots fill out, blocking them automatically.
- **Behavior Analysis** – Detects bot-like behavior (e.g., rapid clicks, unusual mouse movement) instead of requiring user input.
- **reCAPTCHA v3** – Works silently in the background, scoring users based on risk instead of puzzles.
- **Biometric Authentication** – Face recognition, fingerprint scans, or voice verification for secure logins.
- **Email/SMS Verification** – Sends a one-time code to a user's email or phone for validation.
- **Device Fingerprinting** – Identifies unique user devices to detect suspicious logins without user input.