

Software Image Puzzle Using Data Security Messages

Dr.A.Rajesh ,Vilapara Supraja, Yapashetti Varshith,Sreeramoju Akash Tej

Associate Professor, Student ,Student, Student

Department of CSE

Guru Nanak Institute of Technology

adepurajeshadepu@gmail.com , vilaparasupraja@gmail.com , yapashetti.varshith@gmail.com ,
akashte28j@gmail.com

Abstract : The paper “Software Image Puzzle Using Data Security Messages” is designed to provide a secure and interactive method for protecting confidential information using image-based puzzle systems. In the modern digital world, secure communication and data privacy have become major concerns due to cyber threats, unauthorized access, and data leakage. Existing approaches provide limited protection and are vulnerable to attacks such as DoS/DDoS and unauthorized message extraction. To overcome these challenges, the proposed system combines image processing, encryption, steganography, and puzzle generation techniques to ensure secure message transmission.

The system works by dividing an uploaded image into multiple puzzle blocks and randomly shuffling them to create an image puzzle. Secret messages are encrypted and hidden inside the image using secure data hiding techniques. Only authenticated users who successfully solve the puzzle can retrieve and decrypt the hidden information. The application provides secure authentication, puzzle verification, message encryption/decryption, and controlled access mechanisms to maintain data confidentiality and integrity.

Keywords : Image Puzzle, Data Security, Steganography, Encryption, Software Puzzle, Secure Communication, Authentication, GPU-Inflated DoS Attack, Confidential Data Protection.

I. INTRODUCTION

This introduces “Software Image Puzzle Using Data Security Messages,” a secure software application that combines image processing, encryption, steganography, and puzzle generation techniques. The system hides secret messages inside images and converts the images into shuffled puzzle blocks. Only authorized users who correctly solve the puzzle can retrieve the original image and access the hidden information. This approach improves security by adding multiple protection layers including authentication, encryption, and puzzle verification.

The system also focuses on defending against GPU-inflated DoS/DDoS attacks using software puzzle schemes. Unlike traditional client puzzles, software puzzles dynamically generate puzzle algorithms, preventing attackers from preparing GPU-based solutions in advance. This improves the security and reliability of the application while ensuring secure access to protected information.

II. LITERATURE SURVEY

Suriadi, S., Stebila, D., Clark, A., & Liu, H. (2010)

This paper discusses defending web services against Denial of Service (DoS) attacks using client puzzle mechanisms. It explains how cryptographic puzzles can reduce malicious traffic by forcing clients to solve computational problems before accessing services. The study highlights the effectiveness of puzzle-based security in protecting server resources from flooding attacks.

Srivatsay, M., Iyengar, A., Yin, J., & Liu, L. (2012)

This research presents a lightweight and client-transparent method for defending against DoS attacks. The proposed technique operates using browser-based JavaScript mechanisms without requiring additional client software installation. The study improves security while maintaining ease of access for legitimate users.

Smith, F., Grossman, D., Morrisett, G., Hornof, L., & Jim, T. (2005)

This paper explains runtime code generation and dynamic software protection methods. It introduces optimization and code generation techniques that improve software performance and security. The concepts are useful for implementing protected software puzzle systems.

Kaiser, E., & Feng, W. (2007)

The study introduces mod kaPoW, a transparent Proof-of-Work mechanism for mitigating DoS attacks. It focuses on computational challenge systems that prevent malicious clients from overwhelming servers. The research emphasizes adaptive security mechanisms for network protection.

III. METHODOLOGY**A. User Interface Design**

The system provides a secure and user-friendly interface for accessing the application. Users enter their login credentials such as username and password through the login window. The entered details are validated by the server to verify authorized users and prevent unauthorized access. If invalid credentials are entered, the system displays an error message and denies access. The interface is developed using JSP technology to ensure proper authentication and smooth interaction between the user and the server.

B. GPU-Inflated DoS Attack

The system analyzes GPU-inflated Denial of Service (DoS) attacks where malicious users exploit GPU resources to solve puzzles faster than normal CPU processing. When a client requests a service, the server sends a puzzle challenge. Genuine users solve the puzzle using the host CPU and return the response to the server. However, attackers may transfer the challenge to GPU devices to accelerate puzzle solving and overload the server resources.

C. Software Puzzle Generation

The system generates secure software puzzles to defend against automated attacks. The puzzle generation process includes puzzle core generation, puzzle challenge creation, and software puzzle encryption/obfuscation. The server selects instruction blocks using hash functions and secret keys to construct puzzle cores. Additional information such as IP addresses and cookies are used to create challenge messages. These puzzles are dynamically generated to prevent attackers from translating CPU-based puzzles into GPU-compatible versions.

D. Code Protection

The system applies code protection techniques such as code encryption and obfuscation to secure the software puzzle mechanism. Obfuscation increases the complexity of reverse engineering and prevents attackers from understanding the internal working of the puzzle. Instructions and operands within the software code are encrypted to provide temporary but effective protection against unauthorized analysis and modification.

E. Security Analysis

The system performs security analysis to evaluate the effectiveness of software puzzles against GPU-based attacks. The analysis focuses on preventing attackers from solving puzzles through GPU simulation, algorithm cracking, puzzle reproduction, or misuse of access priority. By utilizing different instruction sets and real-time execution environments between CPU and GPU, the system strengthens resistance against malicious activities and improves overall application security.

IV. SYSTEM ARCHITECTURE

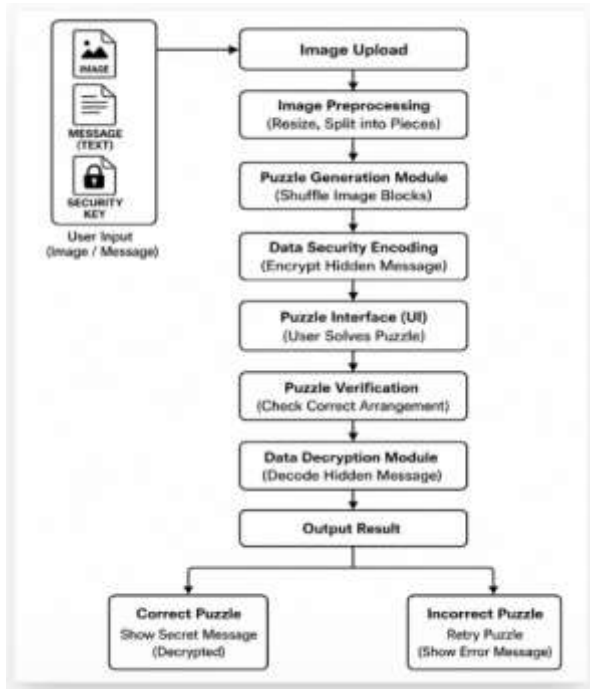


Fig 1 : System Architecture

V. IMPLEMENTATION

The implementation process of the proposed system is carried out through a secure software image puzzle framework using data security messages. The system utilizes software puzzle generation, code protection mechanisms, and secure authentication methods to provide strong protection against unauthorized access and malicious attacks. The framework ensures secure communication between users and the server while maintaining data confidentiality and integrity.

The architecture divides the system into several modules such as user interface design, software puzzle generation, GPU-inflated DoS attack prevention, code protection, and security analysis. This modular structure improves scalability, flexibility, and system performance. Software puzzles are dynamically generated and protected through encryption and obfuscation techniques to prevent attackers from exploiting GPU resources for unauthorized puzzle solving. The system also validates users through secure login authentication developed using Java, JSP, and Servlets with a MySQL database backend.

The architecture further supports secure execution of software puzzles and protection against reverse engineering attacks. It enables smooth interaction between the front-end interface and back-end processing modules while ensuring secure data transfer and storage. Overall, the system provides a secure and efficient environment for protecting software applications using data security messages and software puzzle techniques.

Existing System

Existing security systems mainly depend on traditional authentication and basic encryption techniques to protect software applications and user data. These systems often use static security mechanisms that can easily be bypassed by advanced attackers using GPU acceleration and reverse engineering tools. Traditional systems are vulnerable to Denial of Service (DoS) attacks, unauthorized access, code tampering, and password attacks due to insufficient security measures.

Most existing systems lack dynamic puzzle generation and advanced code protection mechanisms, making them weak against automated attacks. Attackers can exploit GPU resources to solve computational puzzles faster and compromise system security. Existing approaches also provide limited protection against reverse engineering and fail to ensure secure communication between users and servers. As a result, these systems cannot guarantee complete protection of software applications and sensitive data.

Proposed System

The proposed system introduces a secure software image puzzle framework using data security messages to enhance application security and prevent malicious attacks. The framework uses dynamically generated software puzzles, secure authentication, code obfuscation, and encryption techniques to protect the system from unauthorized access and GPU-inflated DoS attacks.

The proposed system generates software puzzles using instruction blocks, hash functions, secret keys, and challenge-response mechanisms. Code protection techniques such as encryption and obfuscation are applied to secure the puzzle logic and increase the difficulty of reverse engineering. The framework also performs security analysis to detect and prevent GPU-based attacks, puzzle cracking attempts, and unauthorized modifications.

The system architecture is designed modularly to independently manage user authentication, puzzle generation, code protection, and security analysis processes. Security features such as encrypted credentials, server authentication, and protected communication channels are incorporated to maintain confidentiality and integrity. Overall, the proposed system provides a reliable, secure, and efficient platform for protecting software applications using software image puzzles and data security messages.

VI. RESULTS

HOME PAGE

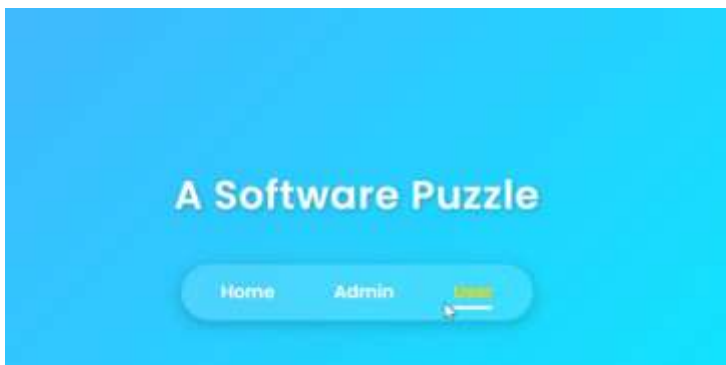


Fig 1: Home Page

This is the main home page of the software puzzle system. It provides navigation options like Home, Admin, and User for accessing different sections of the application.

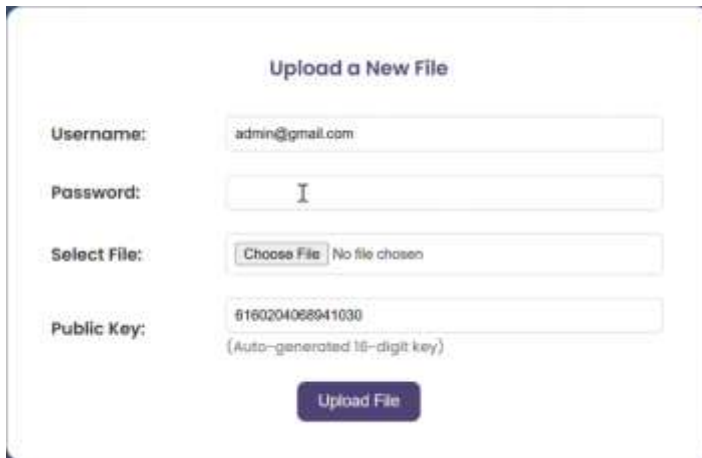
ADMIN PAGE



Fig 2: Admin Page

This page allows the administrator to securely log in using an admin name and password. It also includes Login, Reset, and Back to Home options for easy navigation.

UPLOAD PAGE

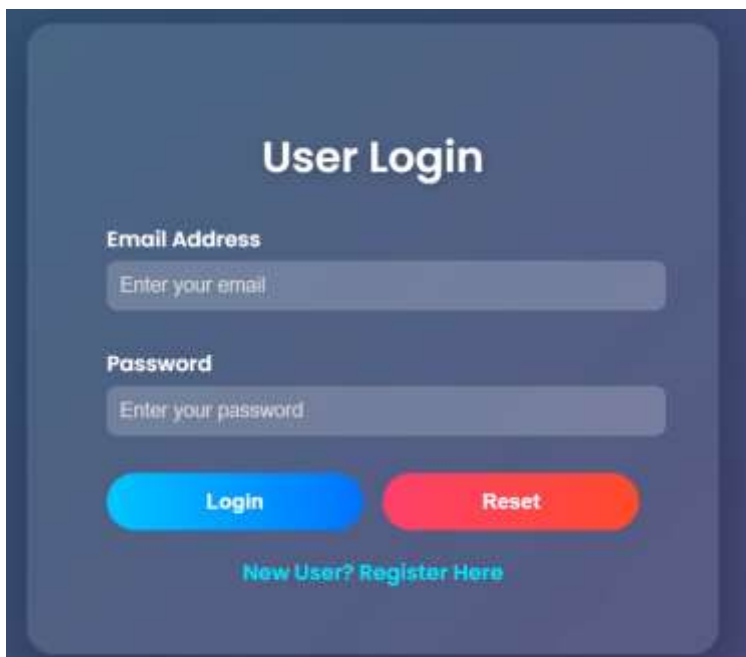


The screenshot shows a web form titled "Upload a New File". It contains four input fields: "Username:" with the value "admin@gmail.com", "Password:" with a cursor, "Select File:" with a "Choose File" button and "No file chosen" text, and "Public Key:" with the value "6160204068941030" and a note "(Auto-generated 16-digit key)". A purple "Upload File" button is at the bottom.

Fig 3 : Upload Page

This page is used by the admin to upload files into the system. It includes fields for username, password, file selection, and an auto-generated public key for security purposes.

USER LOGIN PAGE



The screenshot shows a "User Login" page with a dark blue background. It features two input fields: "Email Address" with the placeholder "Enter your email" and "Password" with the placeholder "Enter your password". Below these are two buttons: a blue "Login" button and a red "Reset" button. At the bottom, there is a link "New User? Register Here" in teal text.

Fig 4: User Login Page

This page enables users to log in with their email address and password. It also provides reset functionality and a registration link for new users.

USER REGISTRATION PAGE

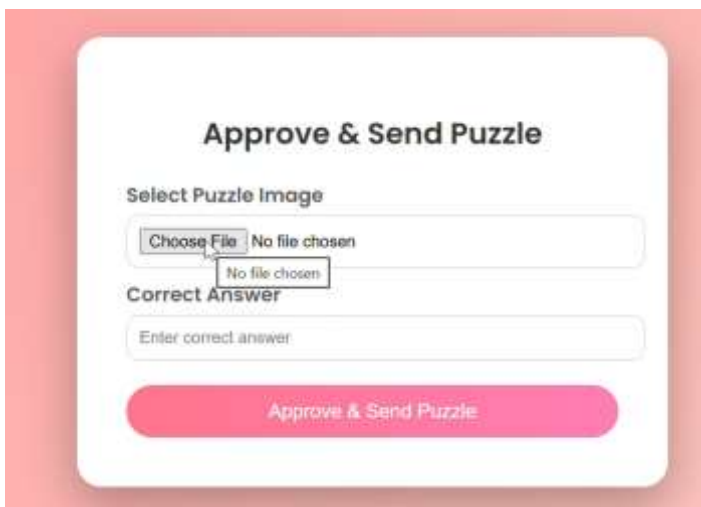


The image shows a user registration form with a purple background. The form includes the following fields: Name (with placeholder text 'Enter your full name'), User ID (with placeholder text '2'), Password (with placeholder text 'Create a strong password'), Mobile (with placeholder text 'Enter your 10-digit mobile number'), Email ID (with placeholder text 'Enter your email address'), and Date (with placeholder text '23/04/2026').

Fig 5:User Registration Page

This image shows the user registration form of the secure file access system. Users can enter details like name, user ID, password, mobile number, email ID, and date to create an account securely.

SEND PUZZLE PAGE



The image shows a 'Send Puzzle' page with a white background and a pink border. The page has the title 'Approve & Send Puzzle'. Below the title, there is a section 'Select Puzzle Image' with a 'Choose File' button and a 'No file chosen' message. Below this, there is a 'Correct Answer' section with a text input field and a 'No file chosen' message. At the bottom, there is a large pink button labeled 'Approve & Send Puzzle'.

Fig 6: Send Puzzle Page

This interface enables the admin to upload a puzzle image and enter the correct answer. After submission, the puzzle is sent to the user as a security verification step before granting file access.

PUZZLE SOLVED PAGE



Fig 7: Puzzle Solved

This screen shows the puzzle verification process for secure authentication. The user must solve the drag-and-drop puzzle correctly to prove authorization and continue file access.

DOWNLOAD PAGE



Fig 8: Download Page

This screen shows the final secure download portal after puzzle verification. The approved file status and download option are displayed, allowing the user to securely download the file.

VII. CONCLUSION

The proposed system for Software Image Puzzle using Data Security Messages provides an effective and reliable solution for protecting software applications against unauthorized access and malicious attacks. By integrating software puzzle generation, code protection mechanisms, encryption techniques, and security analysis, the system ensures strong security, confidentiality, and protection of sensitive data compared to traditional security methods. The use of dynamically generated software puzzles prevents attackers from exploiting GPU resources and performing automated attacks, while code obfuscation and encryption techniques protect the internal logic of the application from reverse engineering and tampering.

The system also supports secure user authentication and protected communication between users and servers, reducing the risk of unauthorized access and data breaches. Advanced security mechanisms such as puzzle-based verification and encrypted data handling improve the overall reliability and efficiency of the system. Overall, this project demonstrates the importance of advanced security technologies in safeguarding software systems, enhancing application protection, and providing a secure environment for data communication through software image puzzles and data security messages.

VIII. FUTURE ENHANCEMENT

The next generation of the Software Image Puzzle using Data Security Messages system will focus on improving security, performance, and resistance against advanced cyberattacks through the use of modern protection technologies and intelligent security mechanisms. The system can be enhanced with stronger encryption algorithms, advanced code obfuscation methods, and optimized software puzzle generation techniques to provide better protection against GPU-based attacks and reverse engineering attempts.

Future improvements may include the integration of Artificial Intelligence and Machine Learning techniques for real-time threat detection, attack prediction, and automated security analysis. The system can also be upgraded with biometric authentication and multi-factor authentication methods to strengthen user verification and access control. In addition, cloud-based deployment and mobile application support can be introduced to improve accessibility, scalability, and remote security management.

REFERENCES

- [1] K. Bilal, S. U. Khan, L. Zhang, H. Li, K. Hayat, S. A. Madani, N. Min-Allah, L. Wang, D. Chen, M. Iqbal, C. Z. Xu, and A. Y. Zomaya, "Quantitative comparisons of the state of the art data center architectures," *Concurrency and Computation: Practice and Experience*, Vol. 25, No. 12, 2013, pp. 1771-1783.
- [2] K. Bilal, M. Manzano, S. U. Khan, E. Calle, K. Li, and A. Zomaya, "On the characterization of the structural robustness of data center networks," *IEEE Transactions on Cloud Computing*, Vol. 1, No. 1, 2013, pp. 64-77.
- [3] D. Boru, D. Kliazovich, F. Granelli, P. Bouvry, and A. Y. Zomaya, "Energy-efficient data replication in cloud computing datacenters," In *IEEE Globecom Workshops*, 2013, pp. 446-451.
- [4] Y. Deswarte, L. Blain, and J-C. Fabre, "Intrusion tolerance in distributed computing systems," In *Proceedings of IEEE Computer Society Symposium on Research in Security and Privacy*, Oakland CA, pp. 110-121, 1991.
- [5] B. Grobauer, T. Walloschek, and E. Stocker, "Understanding cloud computing vulnerabilities," *IEEE Security and Privacy*, Vol. 9, No. 2, 2011, pp. 50-57.