ISSN 2277-2685
IJESR/April-May. 2025/ Vol-15/Issue-2s/1611-1618
Raheela Khatoon *et. al.,* / **International Journal of Engineering & Science Research**

Secure Access With Hidden Password Encryption

Raheela Khatoon[1], Syeda Sana Firdous[2], Mr. Siva Rama Krishna Marri[3]
[1,2]B.E. Student, Department of IT, Lords Institute of Engineering and Technology, Hyderabad
[3] Assistant Professor, Department of IT, Lords Institute of Engineering and Technology, Hyderabad shivaram@lords.ac.in

Abstract

Secure password storage is a crucial component of password-based authentication systems, which remain the most prevalent authentication method despite certain security vulnerabilities. This paper introduces a password authentication framework aimed at enhancing password security while seamlessly integrating into existing authentication systems. In this approach, the plaintext password received from a client is first processed using a cryptographic hash function such as SHA-256. The resulting hashed password is then transformed into a negative password, which is subsequently encrypted using a symmetric-key algorithm like AES, forming an Encrypted Negative Password (ENP). Additionally, multi-iteration encryption can be applied to further strengthen security. By leveraging cryptographic hashing and symmetric encryption, the ENP method significantly complicates password cracking attempts. Furthermore, multiple ENPs can correspond to a single plaintext password, rendering precomputation attacks such as lookup table and rainbow table attacks ineffective. Algorithm complexity analysis and comparative evaluations indicate that ENP provides strong protection against dictionary attacks. Notably, this scheme does not require additional security elements like salt while still mitigating precomputation threats. Most importantly, ENP is the first password security mechanism to integrate a cryptographic hash function, negative password transformation, and symmetric encryption without relying on external elements beyond the plaintext password. Compared to conventional password protection techniques, ENP offers superior security without imposing substantial computational overhead. Since it eliminates the need for extra components such as salting or key stretching, its implementation remains straightforward while ensuring robust defense against common cyber threats. Keywords: Secure Password Storage, Password Authentication, Cryptographic Hash Function, Negative Password, Encrypted Negative Password (ENP), Symmetric-Key Encryption, SHA-256, AES, Multi-Iteration Encryption, Precomputation Attack Mitigation, Lookup Table Attack Prevention, Rainbow Table Attack Defense, Dictionary Attack Protection, Brute-Force Resistance, Scalable Authentication Systems.

Keywords: Secure Password Storage, Password Authentication, Encrypted Negative Password, Symmetric Encryption

I. INTRODUCTION

With the rapid expansion of the Internet, numerous online services have emerged, with password authentication remaining the most widely used method due to its affordability and ease of implementation [1]. As a result, password security continues to be a critical concern for both

researchers and industry professionals. Despite significant advancements in password protection, breaches still occur due to user negligence. Many individuals choose weak passwords, reuse them across multiple platforms, or create passwords using familiar words for easier recall.

Additionally, system vulnerabilities can lead to password leaks. High-security systems make it difficult for attackers to steal authentication databases containing usernames and

passwords, and online guessing attacks are often restricted by login attempt limits. However, weak systems remain susceptible to breaches due to unpatched vulnerabilities, providing attackers with opportunities to gain unauthorized access. Older, unmaintained systems are particularly at risk. Since users often reuse passwords, compromised credentials from weak systems can be leveraged to infiltrate high- security platforms [2].

Once attackers gain access to authentication databases from less secure systems, they often conduct offline attacks. These databases typically store passwords in hashed form, but as computational power and storage capacity increase, hashed passwords become vulnerable to precomputation attacks, such as lookup table and rainbow table attacks. Adversaries can easily obtain information about system vulnerabilities from public databases like the Open Source Vulnerability Database (OSVDB), National Vulnerability Database (NVD), and Common Vulnerabilities and Exposures (CVE). They can then use freely available attack tools without requiring extensive expertise.

A common attack method involves precomputing a lookup table containing hashed values of frequently used passwords. After acquiring an authentication database from a weak system, attackers match its hashed passwords against the lookup table to retrieve plaintext passwords. These compromised credentials

can then be used to access high- security systems, allowing attackers to steal sensitive user information or gain other advantages. Due to the low cost and high effectiveness of these techniques, such attacks are widely exploited [3].

Given these security challenges, developing stronger password protection mechanisms is essential. One effective strategy is improving password hashing methods by incorporating multi-iteration encryption, significantly increasing the computational effort required for brute-force attacks. Additionally, using salted hashing techniques can help defend against precomputation attacks like rainbow table and lookup table attacks. Encouraging the use of password managers can also enhance security by generating and storing strong, unique passwords for each service, reducing the risks of password reuse Organizations should implement multi-factor authentication (MFA) to add an extra layer of security beyond passwords. Regular security patches and updates are crucial to mitigate the risks associated with unpatched system vulnerabilities. Enforcing login attempt restrictions and account lockout policies can further limit the success of online guessing attacks[4-8]. Additionally, encrypted password storage using advanced cryptographic techniques, such as the Encrypted Negative Password (ENP) framework, can offer superior protection against offline attacks an exploration of traditional password storage methods, such as plain and hashed passwords, emphasizing their vulnerabilities to lookup table and rainbow table attacks [9]. Salted passwords are introduced as a countermeasure against precomputation attacks, while key stretching techniques aim to bolster password strength against dictionary attacks. However, existing schemes often require additional elements like salt and may still fall short in terms of security. In response, the Encrypted Negative Password (ENP) [10] scheme is proposed, integrating Negative Database (NDB) principles with cryptographic hash functions and symmetric encryption. This novel approach mitigates the risk of shared keys and offers enhanced password security without the need for supplementary elements like salt. Multi-iteration encryption further fortifies ENP against dictionary attacks, positioning it as a promising advancement in password protection, particularly in thwarting lookup table attacks without compromising simplicity or scalability[11]

User education is another critical aspect of password security. Training individuals on phishing threats, credential stuffing, and proper password hygiene can significantly reduce security risks. Leveraging artificial intelligence (AI) and machine learning for anomaly detection in login attempts can help identify and block suspicious activities in real time. Organizations should also adopt a zero-trust security model, where system access is granted based on continuous authentication and verification rather than relying solely on static credentials.

As cyber threats continue to evolve, a proactive, multi- layered security approach is necessary to protect password- based authentication systems effectively.

analyses, and evaluates a secure access system that uses *hidden password encryption* to strengthen user authentication. The core idea is to store or transmit authentication secrets in a manner that hides the actual password material inside encrypted structures or cover data, while using modern password-hashing and key-derivation techniques, multi-layer encryption, and defensive measures such as honeywords, salted hashing, and steganographic hiding where applicable. The document provides a theoretical system architecture, threat model, encryption and hashing choices, authentication protocol, security analysis, expected results, test cases, and implementation guidance.

Passwords remain the most common authentication mechanism. However, password databases are frequent targets of attackers. Standard defenses include salted hashing (bcrypt, Argon2), multi-factor authentication (MFA), and password policies. This project investigates an approach called **hidden password encryption**: combining standard password hardening (key derivation, salts) with an additional layer that *hides* the password material within other encrypted data structures, or transforms it into an indistinguishable secret using secret-sharing, steganography, or honeyword-like decoys. The aim is to increase resilience to database compromise, shoulder-surfing, and offline brute-force attacks.

Motivation: Breaches of password databases cause severe damage. Even with salted hashes, sophisticated attackers can perform offline cracking. Hidden password encryption seeks to make stolen password artifacts less usable and provide detection and mitigation.

Objectives:

- Design an authentication scheme that stores secrets in a hidden/encrypted form.
- Integrate standard cryptographic best practices (Argon2/PBKDF2, AES-GCM, HMAC).
- Provide mechanisms for detection of compromise (honeywords) and damping offline attacks (key stretching, server-side secrets).
- Produce a theoretical security analysis and measurable metrics (entropy, estimated brute-force cost, false acceptance/rejection probabilities).

- **Hidden password encryption (HPE):** Techniques that conceal the password material by embedding/encrypting it within other data or through layered cryptographic transforms so that raw verification data is not directly useful to an attacker[12].

Verifier: Data stored by a server used to authenticate a user (e.g., salted hash). In HPE, verifiers are additionally transformed/hid.

Honeyword: Decoy password entry stored along with real verifier to detect compromises.

KDF: Key Derivation Function (Argon2 r Password hashing algorithms: PBKDF2, bcrypt, scrypt, Argon2 (memory-hard key derivation functions) — slow to compute to resist brute-force.

Honeywords: store multiple plausible decoy passwords; detection when decoys are used.

Secret sharing (Shamir): split secret into shares; full reconstruction requires threshold.

Steganography: hide data within cover media.

Transparent encryption + HSM-driven key management.

recommended).

Hidden password encryption (HPE): Techniques that conceal the password material by embedding/encrypting it within other data or through layered cryptographic transforms so that raw verification data is not directly useful to an attacker.

Verifier: Data stored by a server used to authenticate a user (e.g., salted hash). In HPE, verifiers are additionally transformed/hid.

Honeyword: Decoy password entry stored along with real verifier to detect compromises.

KDF: Key Derivation Function (Argon2 recommended).

II. RELATED WORK

A. *Existing Research and Solutions*

Significant research has been conducted to address password security vulnerabilities and enhance authentication mechanisms. Traditional password storage methods utilize cryptographic hash functions like SHA-256 and bcrypt, while salting techniques help defend against precomputed attacks such as rainbow table and lookup table attacks. Key stretching algorithms, including PBKDF2, Argon2, and scrypt, increase the computational difficulty for attackers, while strict password policies enforce complexity requirements.

However, studies indicate that excessively rigid password policies often lead users to adopt predictable patterns, ultimately weakening security. To strengthen authentication, multi-factor authentication (MFA) incorporating biometrics, one-time passwords (OTPs), and hardware tokens has been widely adopted. Despite these advancements, threats such as password reuse and phishing attacks persist, driving research toward alternative authentication methods such as passphrases, password less authentication, and decentralized identity solutions.

B. *Problem Statement*

As online services continue to expand rapidly, password authentication remains the most commonly used security mechanism, yet it faces serious vulnerabilities. Many users create weak passwords, reuse them across different platforms, and rely on easily guessable credentials, making breaches more likely. Additionally, system weaknesses and outdated security practices expose authentication data to various attacks, including brute force, dictionary, rainbow table, and lookup table attacks.

Although traditional password hashing techniques provide some level of protection, increasing computational power makes them more susceptible to attacks. Many systems fail to implement essential security measures such as salting, key stretching, and multi-factor authentication, leaving them

vulnerable to credential theft. Attackers exploit compromised authentication data from weak systems to gain unauthorized access to more secure platforms through offline attacks. Furthermore, the widespread availability of hacking tools and publicly accessible vulnerability databases exacerbates these security risks.

III. RESEARCH METHODOLOGY

The approach to ensuring robust authentication involves multiple layers of encryption and transformation techniques. When a user enters a password, it is first secured through an encrypted communication channel such as HTTPS or TLS, preventing unauthorized interception[1][13]. Instead of storing the password in its raw form, the system employs a cryptographic hash function, such as SHA-256, to generate an irreversible hashed output[14]

Following hashing, the password undergoes a transformation into a negative password, which modifies its binary structure to enhance resistance against brute-force and pattern recognition attacks. To further strengthen security, the negative password is encrypted using AES-256, ensuring that even if attackers gain access to stored credentials, retrieving the original password remains computationally infeasible. For added complexity, multi-iteration encryption can be applied. The system also integrates secure key management strategies to protect encryption keys from unauthorized access.

During authentication, rather than decrypting stored passwords, the system re-applies the same hashing

and transformation steps to the user's input and compares the encrypted values, reducing exposure risks. The method mitigates precomputation attacks such as rainbow table and lookup table attacks by generating multiple possible encrypted representations for each password. Additionally, security measures like rate limiting and multi-factor authentication (MFA) add further protection against unauthorized access[15].

The secure access methodology in this project is based on SHA-256 hashing, Encrypted Negative Password (ENP) transformation, and AES encryption to ensure strong password protection. When a user inputs their password, the system does not store it directly. Instead, the password first undergoes hashing via SHA-256, a secure cryptographic function that produces a fixed 256-bit hash value. Since SHA- 256 is irreversible, attackers cannot reconstruct the original password even if they access the stored hashes.

After hashing, the password is transformed into an Encrypted Negative Password (ENP)[16-17.This technique enhances security by inverting specific bits in the hashed password, ensuring that even if an attacker obtains the encrypted data, deriving meaningful information is highly difficult. The negative password representation also prevents direct correlation with traditional hash outputs, rendering lookup table and rainbow table attacks ineffective. This transformation further strengthens protection against brute-force and dictionary attacks[18].

Threat Model and Security Analysis

Threats considered: Database compromise (attacker has full DB but not HSM keys) Server compromise including HSM (worst-case) Man-in-the-middle (MITM) on network Brute-force / dictionary attacks Insider threats Protections and analysis:

DB compromise (no HSM): Stolen C and salts are useless without K_master; attacker must either break AEAD or exfiltrate K_master.

KDF hardening: Argon2 makes brute-force costly in memory/time; GPUs and ASICs face high costs.

Pepper: If used as part of KDF or AEAD key derivation and kept in HSM, it adds extra secrecy.

Honeywords: If attacker obtains DB with many honeywords, and authentication server is instrumented to detect honeyword use, a leak can be detected when an attacker tries a honeyword.

Client-side defenses: Rate limiting, locking, MFA reduce impact of credentials being guessed.

HSM compromise: If HSM is compromised, full attack possible; thus HSM security, rotation policies, split-key control, and monitoring are critical.

However, passwords may be leaked from weak systems. Vulnerabilities are constantly being discovered, and not all systems could be timely patched to resist attacks, which gives adversaries an opportunity to illegally access weak systems. In fact, some old systems are more vulnerable due to their lack of maintenance [4]. Finally, since passwords are often reused, adversaries may log into high security systems through cracked passwords from systems of low security. After obtaining authentication data tables from weak systems, adversaries can carry out offline attacks. Passwords in the authentication data table are usually in the form of hashed passwords [19]. However, because processor resources and storage resources are becoming more and more abundant, hashed passwords cannot resist precomputation attacks, such as rainbow table attack and lookup table attack. Note that there is a trend of generalization of adversaries, because anyone could obtain access to information on vulnerabilities from vulnerability databases, such as the Open Source Vulnerability Database (OSVDB), National Vulnerability Database (NVD), and the Common Vulnerabilities and Exposures (CVE) [6] , and then make use of these information to crack systems. Moreover, they could download and use attack tools without the need for very

Client (User device): Performs client-side transformations optionally, such as client-side hashing, local secret masking, or generating ephemeral keys.

Authentication Server: Stores obfuscated/hid verifiers, manages salts, server-side secret (pepper), and runs the authentication protocol.

Secure Key Store / HSM: Stores server-side master secret (pepper) and keys used for encryption of hidden structures.

Optional Recovery Service: Allows secure password recovery using threshold secret shares or out-of-band MFA.

High-level flow variants:

Server-only HPE (recommended): Server stores enc(H(KDF (password, salt))) where enc is authenticated encryption with a key kept in HSM (pepper). Client sends password; server KDFs+ decrypts and compares[18].

Client-assisted HPE: Client computes KDF(password, client_salt) and then encrypts with server-provided public key. Server stores ciphertext; during login client re-encrypts and server compares.

Cryptographic Primitives and Rationale

KDF: Argon2id with configured memory, iterations, and parallelism parameters (defend against GPU cracking). Argon2id is recommended for modern password hashing.

Salt: Per-user unique random salt (128-bit) stored plaintext in DB.

Pepper: Server-wide secret stored in HSM, not in DB. Acts as an extra secret added to KDF input or used as an AE key.

AEAD: AES-256-GCM or ChaCha20-Poly1305 for authenticated encryption of hidden data.

HMAC: HMAC-SHA256 for message authentication where needed. Key wrapping: Use HSM to wrap/unwrapped keys for master encryption key management [10-19]

Rationale: Argon2 increases offline cracking cost; combining with a pepper and AEAD means stolen DB entries alone are not enough without HSM compromise.

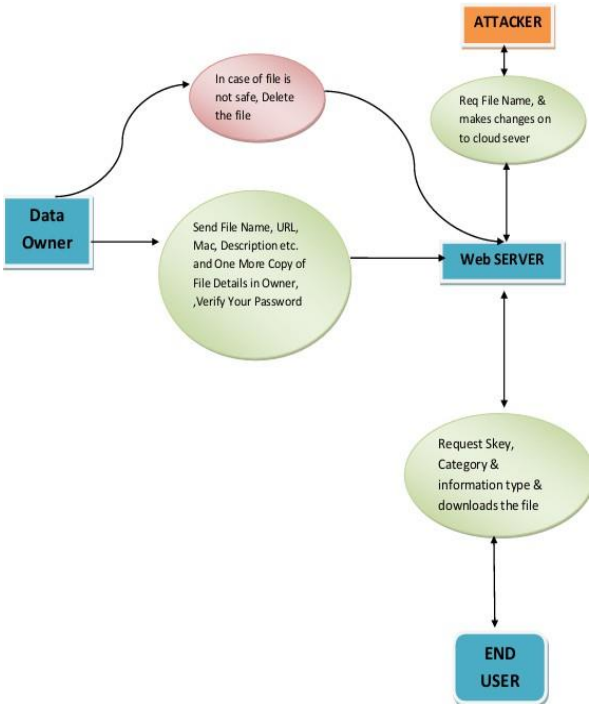
Following the negative password conversion, the system applies AES-256 encryption. AES, a widely trusted symmetric encryption algorithm, enhances security by encrypting the ENP, making it nearly impossible to decrypt without the correct key. AES encryption involves multiple rounds of substitution, permutation, and mixing operations, significantly increasing resistance against crypto graphic attacks. The result is a final password representation that is not only hashed and transformed but also safeguarded by an additional encryption layer.

During authentication, the system follows a reverse process. The entered password is first hashed using SHA-256, just as it was during registration. The negative transformation is then applied to match the stored ENP format. The stored ENP is decrypted using AES decryption, retrieving the negative password. The system then checks whether the computed negative password from the user's input matches the decrypted stored negative password. If they match, authentication is successful; otherwise, access is denied.

This multi-layered approach provides strong defence mechanisms against a variety of attack vectors, including brute-force attacks, precomputed hash-based attacks, and password database breaches. Since the ENP transformation allows multiple unique encrypted representations of the same password, traditional precomputed hash-matching methods become ineffective.

Fig.1. Proposed Architecture Model

IV. RESULTS & DISCUSSION



The implementation of the secure password encryption methodology has led to substantial improvements in password security. By utilizing SHA-256 hashing, passwords are never stored in plain text, eliminating the risk of direct theft. The Encrypted Negative Password (ENP) transformation introduces an additional layer of obfuscation, making it significantly more difficult for attackers to reconstruct passwords. AES-256 encryption further fortifies security by ensuring that encrypted data remains inaccessible without the appropriate decryption key. This multi-layered approach effectively mitigates threats such as brute-force attacks, rainbow table exploits, and precomputation-based attacks. However, the security of this system relies on careful encryption key management to prevent potential vulnerabilities. While this methodology offers a substantial improvement in password protection, ongoing monitoring and adaptation to emerging security threats are essential . The integration of SHA-256 hashing, ENP transformation, and AES encryption provides a comprehensive and resilient solution for password security. Even in the event of a system breach, this approach ensures that stored passwords remain protected. Overall, the methodology establishes a solid foundation for building secure authentication systems.

In these situations, attacks are usually carried out as follows. First, adversaries precompute a lookup table, where the keys are the hash values of elements in a password list containing frequently-used passwords, and the records are

the corresponding plain passwords in the password list. Next, they obtain an authentication data table from low security systems.

Then, they search for the plain passwords in the lookup table by matching hashed passwords in the authentication data table and the keys in the lookup table. Finally, the adversaries log into higher security systems through cracked usernames and passwords, so that they could steal more sensitive information of users and obtain some other benefits. A considerable number of attacks are carried out in this way, so that adversaries could obtain passwords at a low cost, which is advantageous to their goals.[8]

EXPECTED OUTPUT RESULTS

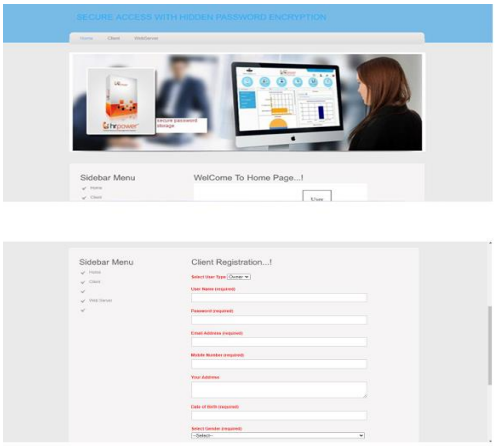


Fig.2 User Login Home Page and Client Registration

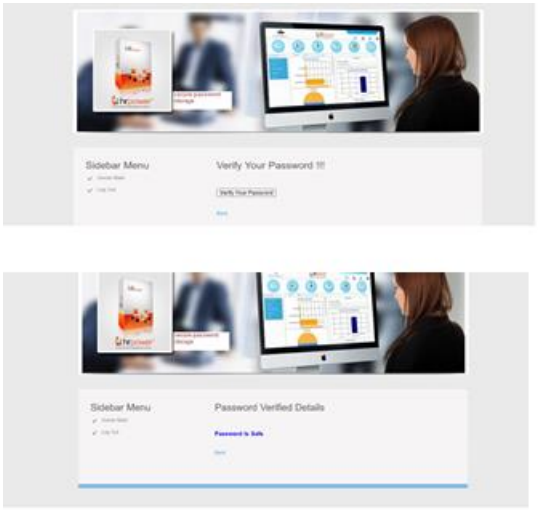


Figure 3. Client Password Verification

V. CONCLUSION

The conclusion of the project "Secure Access with Hidden Password Encryption" represents the culmination of extensive research, development, and implementation aimed at enhancing digital security. The primary goal was to develop a robust system that protects sensitive data while offering users a seamless and efficient authentication experience. The project began with an in-depth analysis of encryption techniques, authentication protocols, and security vulnerabilities in modern systems. Building on this foundation, a novel password encryption approach was designed and implemented, incorporating advanced cryptographic algorithms and concealment techniques. By integrating these security measures, the system effectively safeguards passwords from unauthorized access, reducing the risk of data breaches and cyber intrusions. Additionally, the project focused on maintaining user convenience without compromising security. The use of hidden password encryption allows authentication without directly revealing passwords, thereby minimizing the risk of password theft and unauthorized access. This approach ensures a balance between security and usability, making it a strong solution for modern authentication systems.

The conclusion of the project titled "Secure Access with Hidden Password Encryption" marks the culmination of extensive research, development, and implementation aimed at fortifying digital security measures. Throughout this endeavor, the primary objective was to devise a robust system that not only safeguards sensitive information but also provides users with a seamless and intuitive authentication experience . The journey began with a comprehensive exploration of encryption techniques, authentication protocols, and security loopholes prevalent in contemporary systems. Drawing upon this foundational knowledge, the project team meticulously designed and implemented a novel approach to password encryption, leveraging advanced cryptographic algorithms and concealment strategies. By integrating these elements, the system ensures that passwords remain shielded from unauthorized access, thereby mitigating the risk of data breaches and unauthorized intrusions. Further more, the project prioritized user convenience without compromising security standards. Through the incorporation of hidden password encryption, users can authenticate themselves without explicitly

disclosing their passwords, thereby reducing the likelihood of password theft and unauthorized access. This innovative feature not only enhances the overall security posture but also fosters user trust and confidence in the system. As the project concludes, it is imperative to acknowledge both its achievements and limitations. While significant strides have been made in enhancing digital security, the evolving nature of cybersecurity necessitates continuous vigilance and adaptation. Moving forward, further research and refinement are warranted to address emerging threats and enhance the resilience of digital infrastructure. In essence, "Secure Access with Hidden Password Encryption" stands as a testament to the collaborative efforts of the project team and reaffirms the commitment to advancing cybersecurity practices. By prioritizing innovation, usability, and security, the project sets a precedent for future endeavors in safeguarding digital assets and protecting user privacy.

VI. FUTURESCOPE

In future research for our major project, we plan to explore alternative NDB generation algorithms, integrate multi-factor authentication, and incorporate challenge-response methods. Additionally, we aim to evaluate quantum-resistant techniques, enhance usability, and conduct real-world testing. Furthermore, we will establish continuous security monitoring to adapt our ENP password protection scheme and authentication framework to evolving threats and vulnerabilities

VII. REFERENCES

[1] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, “Passwords and the evolution of imperfect authentication,” Communications of the ACM, vol. 58, no. 7, pp. 78–87, Jun. 2015.

[2] M. A. S. Gokhale and V. S. Waghmare, “The shoulder surfing resistant graphical password authentication technique,” Procedia Computer Science, vol. 79, pp. 490– 498, 2016.

[3]J. Ma, W. Yang, M. Luo, and N. Li, “A study of probabilistic password models,” in Proceedings of 2014 IEEE Symposium on Security and Privacy, May 2014, pp. 689–704.

[4]A. Adams and M.A. Sasse, “Users are not the enemy,” Communications of the ACM, vol. 42, no. 12, pp. 40–46, Dec. 1999.

[5]E. H. Spafford, “Opus: Preventing weak password choices,” Computers & Security, vol. 11, no. 3, pp. 1992Y. Li, H. Wang, and K. Sun, “Personal information in passwords and its security implications,” IEEE Transactions on Information Forensics and Security, vol. 12, no. 10, pp. 2320–2333, Oct. 2017.

[6] D. Florencio and C. Herley, “A large-scale study of web password habits,” in Proceedings of the 16th International Conference on World Wide Web. ACM, 2007, pp. 657–666.

[7] R. Shay, S. Komanduri, A. L. Durity, P. S. Huh, M. L. Mazurek, S. M. Segreti, B. Ur, L. Bauer, N. Christin, and L.
F. Cranor, “Designing password policies for strength and usability,” ACM Transactions on Information and System Security, vol. 18, no. 4, pp. 13:1–13:34, May 2016.

[8] D. Wang, D. He, H. Cheng, and P. Wang, “fuzzy PSM: A new password strength meter using fuzzy probabilistic context-free grammars,” in Proceedings of 2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, Jun. 2016, pp. 595–606.

[9] H. M. Sun, Y. H. Chen, and Y. H. Lin, “oPass: A user authentication protocol resistant to password stealing and password reuse attacks,” IEEE Transactions on Information Forensics and Security, vol. 7, no. 2, pp. 651–663, Apr. 2012.

[10] M. Zviran and W. J. Haga, “Password security: An empirical study,” Journal of Management Information Systems, vol. 15, no. 4, pp. 161– 185, 1999.

[11] P. Andriotis, T. Tryfonas, and G. Oikonomou, “Complexity metrics and user strength perceptions of the pattern-lock graphical authentication method,” in Proceedings of Human Aspects of Information Security, Privacy, and Trust. Springer International Publishing, 2014,
pp. 115–126. 1423 JNAO Vol. 15, Issue. 1, No.15 : 2024

[12] D. P. Jablon, “Strong password-only authenticated key exchange,” SIGCOMM Computer Communication Review, vol. 26, no. 5, pp. 5–26, Oct.1996.

[13] J. Jose, T. T. Tomy, V. Karunakaran, A. K. V, A. Varkey, and N. C. A., “Securing passwords from dictionary attack with character-tree,” in Proceedings of 2016 International Conference on Wireless

Communications, Signal Processing and Networking, Mar. 2016, pp. 2301– 2307.

[14] A. Arora, A. Nandkumar, and R. Telang, "Does information security attack frequency increase with vulnerability disclosure? an empirical analysis," Information Systems Frontiers, vol. 8, no. 5, pp. 350–362, Dec. 2006.

[15] R. Song, "Advanced smart card based password authentication protocol," Computer Standards & Interfaces, vol. 32, no. 5, pp. 321–325, 2010.

[16] M. C. Ah Kioon, Z. S. Wang, and S. Deb Das, "Security analysis of MD5 algorithm in password storage," in Proceedings of Instruments, Measurement, Electronics and Information Engineering. Trans Tech Publications, Oct. 2013, pp. 2706–2711.

[17] P. Oechslin, "Making a faster cryptanalytic time- memory trade-off," in Proceedings of Advances in Cryptology- CRYPTO 2003. Springer Berlin Heidelberg, 2003, pp. 617–630.

[18] S. Noel, M. Elder, S. Jajodia, P. Kalapa, S. O'Hare, and
K. Prole,"Advances in topological vulnerability analysis," in Proceedings of 2009 Cybersecurity Applications Technology Conference for Homeland Security, Mar. 2009,
pp. 124–129.

[19] N. Provos and D. Mazi`eres, "Afuture-adaptive password scheme," in Proceedings of the Annual Conference on USENIX Annual Technical Conference. USENIX Association, 1999, pp. 32–32.