

SECURE ACCESS WITH HIDDEN PASSWORD ENCRYPTION

P.Shailaja, Associate Professor CSE, Vaagdevi College of Engineering (Autonomous), India

B. Mahalaxmi, UG Student, CSE, Vaagdevi College of Engineering (Autonomous), India

A. Nithin, UG Student, CSE, Vaagdevi College of Engineering (Autonomous), India

A. Koushik Raj, UG Student, CSE, Vaagdevi College of Engineering (Autonomous), India

B. Himasri, UG Student, CSE, Vaagdevi College of Engineering (Autonomous), India

ABSTRACT

Secure password storage is a vital aspect in systems based on password authentication, which is still the most widely used authentication technique, despite its some security flaws. In this paper, we propose a password authentication framework that is designed for secure password storage and could be easily integrated into existing authentication systems. In our framework, first, the received plain password from a client is hashed through a cryptographic hash function (e.g., SHA-256). Then, the hashed password is converted into a negative password. Finally, the negative password is encrypted into an Encrypted Negative Password (abbreviated as ENP) using a symmetric-key algorithm (e.g., AES), and multi-iteration encryption could be employed to further improve security. The cryptographic hash function and symmetric encryption make it difficult to crack passwords from ENPs. Moreover, there are lots of corresponding ENPs for a given plain password, which makes precomputation attacks (e.g., lookup table attack and rainbow table attack) infeasible. The algorithm complexity analyses and comparisons show that the ENP could resist lookup table attack and provide stronger password protection under dictionary attack. It is worth mentioning that the ENP does not introduce extra elements (e.g., salt); besides this, the ENP could still resist precomputation attacks. Most importantly, the ENP is the first password protection scheme that combines the cryptographic hash function, the negative password and the symmetric-key algorithm, without the need for additional information except the plain password.

1 INTRODUCTION

OWING to the development of the Internet, a vast number of online services have emerged, in which password authentication is the most widely used authentication technique, for it is available at a low cost and easy to deploy[1]. Hence, password security always attracts great interest from academia and industry [2]. Despite great research achievements on password security, passwords are still cracked since users' careless behaviors. For instance, many users often select weak passwords; they tend to reuse same passwords in different systems; they usually set their passwords using familiar vocabulary for its convenience to remember. In addition, system problems may cause password compromises. It is very difficult to obtain passwords from high security systems. On the one hand, stealing authentication data tables (containing usernames and passwords) in high security systems is difficult. On the other hand, when carrying out an online guessing attack, there is usually a limit to the number of login attempts.[3]

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[5]. 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 professional security knowledge. Some powerful attack tools, such as hashcat, RainbowCrack and John the Ripper, provide a variety of functions, such as multiple hash algorithms, multiple attack models, multiple operating systems, and multiple platforms, which raises a higher demand for secure password storage.[7]

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]

2.LITERATURE SURVEY

The literature survey encompasses 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.

3. PROBLEM STATEMENT

Hashed Password: The simplest scheme to store passwords is to directly store plain passwords. However, this scheme presents a problem that once adversaries obtain the authentication data table, all passwords are immediately compromised. To safely store passwords, a common scheme is to hash passwords using a cryptographic hash function[11], because it is infeasible to directly recover plain passwords from hashed passwords. The cryptographic hash function quickly maps data of arbitrary size to a fixed-size sequence of bits. In the authentication system using the hashed password scheme, only hashed passwords are stored. However, hashed passwords cannot resist lookup table attack. Furthermore, rainbow table attack is more practical for its space-time tradeoff. Processor resources and storage resources are becoming richer, which makes the precomputed tables used in the above two attacks sufficiently large, so that adversaries could obtain a higher success rate of cracking hashed passwords.[12]

Salted Password: To resist precomputation attacks, the most common scheme is salted password. In this scheme, the concatenation of a plain password and a random data (called salt) is hashed through a cryptographic hash function[13]. The salt is usually generated at random, which ensures that the hash values of the same plain passwords are almost always different. The greater the size of the salt

is, the higher the password security is. However, under dictionary attack, salted passwords are still weak. Note that compared with salted password, the ENP proposed in this paper guarantees the diversity of passwords without the need for extra elements (e.g., salt).

Key Stretching: To resist dictionary attack, key stretching [38], which converts weak passwords to enhanced passwords, was proposed. Key stretching could increase the time cost required to every password attempt, so that the power of defending against dictionary attack is increased [14]. In the ENP proposed in this paper[15], like key stretching, multi-iteration encryption is used to further improve password security under dictionary attack[16], and compared with key stretching, the ENP does not introduce extra elements (e.g., salt).

3.1 LIMITATION OF SYSTEMS

System is not secured due to lack of improved dynamic Key-Hashed Message Authentication Code function[17] (abbreviated as d-HMAC). Password protection scheme called Encrypted Negative Password is absent.

4. IMPLEMENTATION

4.1 Home: Tackles the challenge of safeguarding data. We'll explore methods to encrypt passwords, rendering them invisible to unauthorized users[18]. By implementing these techniques, access will be strictly controlled, ensuring only those with the key can unlock the protected information.

4.2 Client: This project aims to develop a secure access system that utilizes hidden password encryption[19]. Passwords will be obscured during storage and transmission, enhancing protection against unauthorized access[20]. The system will likely involve a combination of one-way encryption algorithms and secure key management techniques to safeguard your data [21].

4.3 Web Server: secures your web server by replacing traditional password storage[22]. When a user logs in, their password is never revealed – instead, it's encrypted using a one-way function. This hidden password protects your system from breaches where attackers steal password databases[23].

5. EXPECTED OUTPUT RESULTS



Client Registration...!

Select User Type **Owner**

User Name (required)

Password (required)

Email Address (required)

Mobile Number (required)

Your Address


Date of Birth (required)

Select Gender (required)

--Select--

Registered Successfully

[Home](#)



Client Login...!

User Name:

Password:

Select User Type **Owner**

[Login](#) [Reset](#)

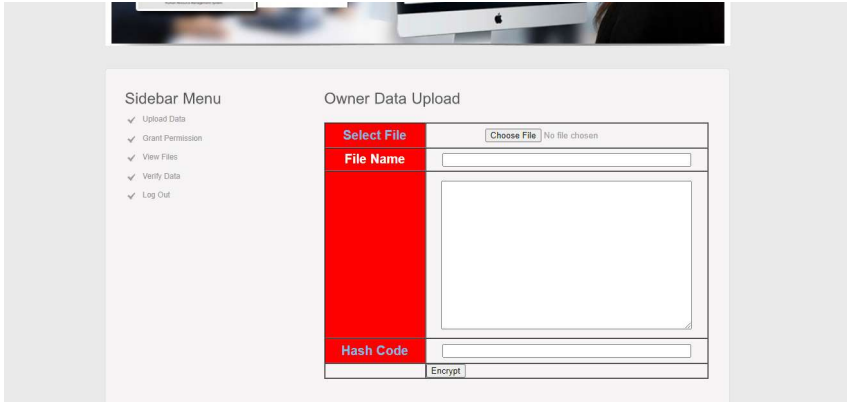
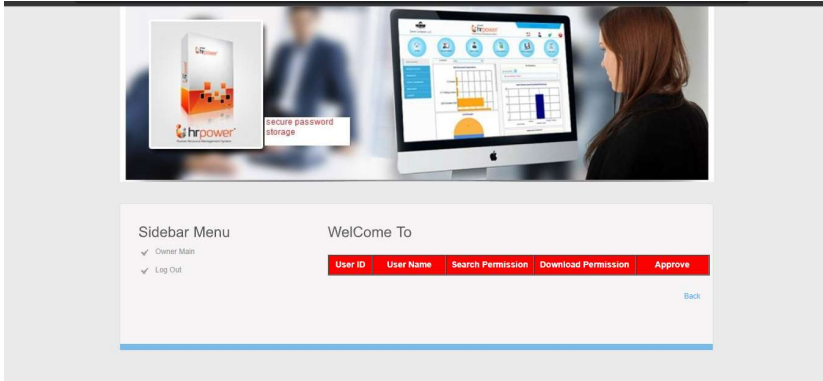
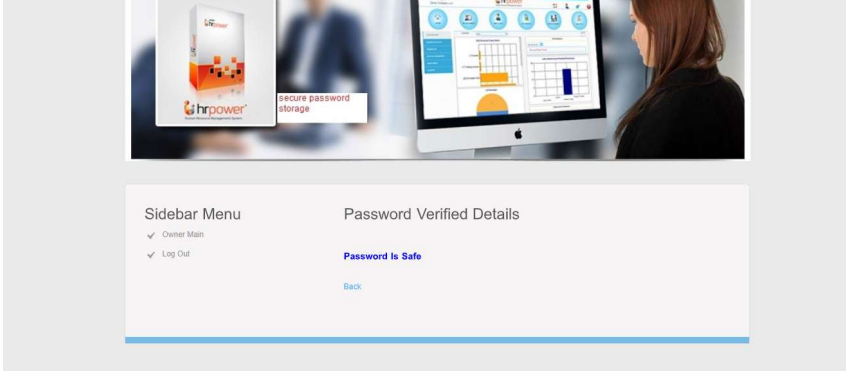
[Back](#)

[New Register](#)



Verify Your Password !!!

[Back](#)



Sidebar Menu

- ✓ Upload Data
- ✓ Grant Permission
- ✓ View Files
- ✓ Verify Your Data
- ✓ Verify Your Password
- ✓ Log Out

WelCome:: venkat
 Your Decrypted ENP::123

```

            graph TD
            User1[User] -- plain password --> Hash[Hash]
            AuthTable[Authentication Data Table] --> Hash
            Hash -- hashed password (key) --> Decrypt[Decrypt]
            ENP[ENP (data)] --> Decrypt
            Decrypt -- negative password and hashed password --> Solution{is solution?}
            Solution -- accept or reject request --> User2[User]
            
```

Sidebar Menu

- ✓ Server Main
- ✓ Log Out

View File_Score

File Type	Score
Connect.jsp	6
KeyGen.java	2
SQL.txt	0
Android.txt	2
Dotnet.txt	0
Java.txt	3
Silverlight.txt	0

[Back](#)

Sidebar Menu

- ✓ Home Page

Attacker

File No	Attacker	File Name	Date & Time	URL & Hash Code	
8	Hacker	Manjunath	Java.txt	05/08/2019 16:14:51	Details


[Back](#)

Sidebar Menu

- ✓ Server Main
- ✓ Log Out

All Clients And Transactions...

SI NO	User Name	File Name	Secret Key	Operation	Date & Time
4	Umesh	Connect.jsp	[B@1f4fedf	Upload	05/08/2019 15:21:02
5	Ramesh	Connect.jsp	[B@1f4fedf	Download	05/08/2019 15:24:39
6	Umesh	KeyGen.java	[B@27aa0	Upload	05/08/2019 15:26:02
7	Umesh	SQL.txt	[B@5d9072	Upload	05/08/2019 15:26:33
8	Manjunath	Android.txt	[B@1d0eefb	Upload	05/08/2019 16:07:27
9	Manjunath	Dotnet.txt	[B@84322	Upload	05/08/2019 16:07:41
10	Manjunath	Java.txt	[B@1b0cc8c	Upload	05/08/2019 16:07:52
11	Manjunath	Silverlight.txt	[B@9955ab	Upload	05/08/2019 16:08:06
12	tmksmanju	Connect.jsp	[B@1f4fedf	Download	05/08/2019 16:10:47
13	tmksmanju	Java.txt	[B@1b0cc8c	Download	05/08/2019 16:12:16
14	tmksmanju	Android.txt	[B@1d0eefb	Download	05/08/2019 16:12:32



Sidebar Menu

- ✓ Server Main
- ✓ Log Out


Permission To User ...

Select Data User Ramesh ▼

Search Access Yes ▼

Download Access Yes ▼

[Back](#)



Sidebar Menu

- ✓ Server Main
- ✓ Log Out

All s...

SN	Owner Name	File Name	Rank	Date & Time	URL and Hash Code
3	Umesh	Connect.jsp	6	05/08/2019 15:21:02	Details
4	Umesh	KeyGen.java	2	05/08/2019 15:26:02	Details
5	Umesh	SQL.txt	0	05/08/2019 15:26:33	Details
6	Manjunath	Android.txt	2	05/08/2019 16:07:27	Details
7	Manjunath	Dotnet.txt	0	05/08/2019 16:07:41	Details
8	Manjunath	Java.txt	3	05/08/2019 16:07:52	Details
9	Manjunath	Silverlight.txt	0	05/08/2019 16:08:06	Details



Sidebar Menu

- ✓ Server Main
- ✓ Log Out


All Clients...

ID	UserImage	Username	Password	Password's Hash Code	Address	Gender	Status	User Type
6		Kumar	QDY3ODE2OHF	-21b4494a446726145208d aa6a000e2356cbf92ef	#7827.4th Cross.Rajajanager	Male	Authorized	Owner
7		Ramesh	UmFIZXNo	-67be3b21e2ecf18cc3aa 76e60bf3986ee76bdf9	#7827.4th Cross.Vijayanagar	Male	Authorized	User
8		Umesh	VW1ic2g=	274d64548be795d51f926d ecbb4e806210f2a10c	#782.4th Cross.Rajajanager	Male	Authorized	Owner

Sidebar Menu

- ✓ View All Users
- ✓ View All Owner Files
- ✓ Give Privileges To Users
- ✓ View Transactions
- ✓ View All Encrypted Negative Password
- ✓ View All Attacker
- ✓ View All Password Attackers
- ✓ View File Score Results
- ✓ Log Out

WelCome To Server Main...!




In this paper, we propose a password authentication framework that is designed for secure password storage and could be easily integrated into existing authentication systems. In our framework, first, the received plain password from a client is hashed through a cryptographic hash function (e.g., SHA-256). Then, the hashed password is converted into a negative password. Finally, the negative password is encrypted into an Encrypted Negative Password (abbreviated as ENP) using a symmetric-key algorithm (e.g., AES), and multi-iteration encryption could be employed to further improve security. The cryptographic hash function and symmetric encryption make it difficult to crack passwords from ENPs.



Sidebar Menu

- ✓ Log Out

UnAuthorized venkat



```

graph TD
    User --> Login
    Login --> Hash
    Hash --> HashedPassword[Hashed Password]
    HashedPassword --> Decrypt
    Decrypt --> IsValid{Is Valid?}
    IsValid --> User
    
```


Registered Successfully

[Home](#)

Client Registration...!

Select User Type **Owner** ▼

User Name (required)

Password (required)

Email Address (required)

Mobile Number (required)

Your Address

Date of Birth (required)

Select Gender (required) **--Select--** ▼



Client Login...!

User Name:

Password:

Select User Type **Owner** ▼

[Login](#) [Reset](#)

[Back](#)

[New Register](#)

SECURE ACCESS WITH HIDDEN PASSWORD ENCRYPTION

[Home](#) [Client](#) [WebServer](#)

WelCome To Home Page...!

User:

CONCLUSION

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 [24]. 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. Furthermore, 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.

FUTURE SCOPE

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.

7. 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. 273–278, 1992.
- [6] Y. 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.
- [7] 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.

- [8] 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.
- [9] D. Wang, D. He, H. Cheng, and P. Wang, "fuzzyPSM: 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.
- [10] 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.
- [11] M. Zviran and W. J. Haga, "Password security: An empirical study," *Journal of Management Information Systems*, vol. 15, no. 4, pp. 161–185, 1999.
- [12] 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.
- [13] D. P. Jablon, "Strong password-only authenticated key exchange," *SIGCOMM Computer Communication Review*, vol. 26, no. 5, pp. 5–26, Oct. 1996.
- [14] 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.
- [15] 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.
- [16] R. Song, "Advanced smart card based password authentication protocol," *Computer Standards & Interfaces*, vol. 32, no. 5, pp. 321–325, 2010.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] "Hashcat," <https://hashcat.net/hashcat/>.
- [21] "RainbowCrack," <http://project-rainbowcrack.com/>.
- [22] "John the Ripper," <http://www.openwall.com/john/>.
- [23] N. Provos and D. Mazières, "A future-adaptive password scheme," in *Proceedings of the Annual Conference on USENIX Annual Technical Conference*. USENIX Association, 1999, pp. 32–32.
- [24] "RFC 7914: The scrypt Password-Based Key Derivation Function," <https://tools.ietf.org/html/rfc7914>.