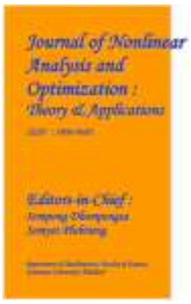
Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 15, Issue. 1 : 2024
ISSN : **1906-9685**



QUICK AND SECURE ONLINE AUTHENTICATION WITH HPAKE HONEY PASSWORD-AUTHENTICATED KEY EXCHANGE

#1 Mrs.K.Harika , #2 S.Aparna, #3 A. Udaya Bhanu, #4 A.Harini, #5 M.Anil Kumar
#1Associative professor in Department of IT, DVR & Dr.HS MIC College of Technology,Kanchikacherla
#2#3#4#5 B.Tech with Specialization of Information Technology , DVR & Dr.HS MIC College of Technology,Kanchikacherla-521180

Abstract_ One of the most widely used safe mechanisms for online applications in the real world is password-only authentication. However, it is easily vulnerable to a real threat: password leaks caused by both internal and external attackers.
Insiders may purposefully steal credentials or unintentionally leak them, while external attackers may compromise the password file kept on the authentication server.
As of right now, there are two primary methods for dealing with the leak: the use of honeywords for external attackers and augmented password-authentication key exchange (aPAKE) for insiders. But none of them are able to withstand both blows.
In order to bridge the gap, we introduce the concept of honey PAKE (HPAKE), which enables the authentication server to identify password leaks and provide security that goes beyond the limitations of aPAKE.

1.INTRODUCTION

The username/secret key worldview is the most usually involved verification system in security applications [3]. Elective validation factors, including tokens and biometrics, require buying extra hard-product, which is frequently thought to be excessively costly for an application. Notwithstanding, passwords are low-entropy insider facts, and dependent upon word reference assaults [3]. Consequently, they should be safeguarded during transmission. The broadly conveyed strategy is to send passwords through SSL/TLS [36]. Yet, this requires Public Key Framework (PKI) set up; it is costly to keep a PKI. Moreover, utilizing SSL/TLS is likely to man-in-the-center assaults [3]. On the off chance that a client confirms himself to a phishing site by unveiling his secret key, the secret word will be taken despite the fact that the meeting is completely encoded. Since passwords are intrinsically feeble, one rationale arrangement appears to supplant them with solid insider facts, say, cryptograph-frigidly secure confidential keys. This approach was embraced by the UK Public Framework Administration (NGS) to verify clients [4]. In the UK, anybody who applies to get to the public framework figuring asset must first create a private/public key sets of his own, and afterward have the public key certified by NGS. The validation strategy for the matrix figuring conditions in the USA is comparable [18]. In any case, improvements in the beyond a decade uncover that clients - the greater part of them are non-PC subject matter experts - experience serious difficulties in dealing with their confidential keys and certificates [5]. This has incredibly thwarted the more extensive acknowledgment of the matrix processing innovation. Consequently, powerless passwords are only an unavoidable truth that we should confront. Scientists have been effectively investigating ways of performing secret key based verification without utilizing PKIs or certificates - an exploration subject called the Secret key Confirmed Key Trade (PAKE) [9]. The first achievement came in

1992 when Beloin and Merrit presented the Squeeze convention [13]. Regardless of a few detailed shortcomings [22, 26, 29, 32], the Squeeze convention first exhibited that the PAKE issue was essentially reasonable. From that point forward, various conventions have been proposed. Large numbers of them are just variations of Squeeze, starting up the "symmetric code" in different ways [9]. The couple of methods that case to oppose realized assaults have practically completely been licensed - most eminently, Squeeze was protected by Bright Advances [15], and SPEKE by Phoenix Innovations [24]. Accordingly, the scientific local area and the more extensive security industry can't promptly benefit from the executions of these procedures [16].The security with the Squeeze and SPEKE conventions is just heuristic. Given how the two methods were planned, formal security confirmations appears to be improbable without presenting new presumptions or loosening up prerequisites; we will make sense of the subtleties in Segment 4. In the accompanying segment, we will acquaint a different approach with tackle thecae issue, and show that our answer is liberated from the security issues revealed with the Squeeze and SPEKE conventions

2.LITERATURE SURVEY

Against Insiders. In Figure 1a, Expanded passwordauthentication key trade (aPAKE) [9] is intended to permit a client and a server to lay out a meeting key in light of a secret key, where the client has the secret phrase plaintext and the server just holds the verifier. This method keeps the server from knowing the secret key, and hence opposes the insider assaults. Since Bellovin and Merrit [9] presented this idea, numerous specialists proposed different aPAKE plans [10], [11], [12], [13] to work on the security and productivity execution. Among them, Hazy [12] is the most all around concentrated on conspire with the most grounded security and accordingly, it as of late is normalized by the Crypto Discussion Exploration Gathering of the Web Designing Team (IETF) [14]. Against Pariahs. Honeyword method [15] (see Figure 1b) is proposed to identify the secret phrase spillage for the most well-known secret phrase just validation frameworks, passwordover-TLS. This approach partners t−1 distraction and plausiblelooking passwords (i.e., honeywords) to each record. The honeywords and the genuine secret key are aggregately called sweet words. On the off chance that an assailant takes the secret phrase record, she can't tell the genuine one and most likely (with 1−1/t likelihood) sign in with a honeyword. Then, at that point, the server can identify the secret phrase spillage from "some unacceptable" login. The subsequent works center around the honeyword age calculations [16], [17] in order to create more conceivable looking imitations and the recognition strategies [18] to further develop dependability. Others. Secret phrase less confirmation [19] or multifaceted validation frameworks [20], [21] take full advantage of different elements, e.g., cell phone and unique mark. They altogether decrease the gamble of secret phrase spillage. On the off chance that an assailant takes the secret phrase, she actually needs extra factors to think twice about. Furthermore, in a portion of these plans, validation server doesn't have to store the secret word related information, so that regardless of whether the aggressor compromises the capacity record on server, she can't do disconnected secret word speculating as long as different variables are secure. An ordinary plan should be visible in [21], [22], [23] that a brilliant gadget (as a confirmation factor) is utilized to store the secret phrase related information, making frameworks oppose disconnected speculating on account of server split the difference. Shortcomings. The strategies above, tragically, have the accompanying inadequacies. The honeyword component requires the client to send the secret word plaintext to the server (through a server-verified secure channel), generally the server can't determine whether the login secret word is genuine. In this way an insider can straightforwardly take the plaintext of the login secret key with next to no speculating assaults. In aPAKE, the server needs to store the verifiers in the secret word document for validation. Yet, an outside aggressor might take the document and complete speculating assaults [24] to recuperate the secret word. This weakness is intrinsic in aPAKE. Furthermore, neither of these techniques can give an answer keeping up with protection from the two insiders and pariahs. With respect to other (passwordless or multifaceted) approaches, they might give more grounded security depending on additional variables, which might carry inconveniences to deployability and ease of use. In this paper, we don't consider them and just spotlight on passwordonly confirmation. As indicated by the above conversation, we in this way bring up an issue: " How is it that one could plan a quick and secure secret key just confirmation conspire that can oppose both the insider and outer aggressors.

3.PROPOSED SYSTEM

In order to provide security beyond the conventional bounds of aPAKE, we introduce the concept of honey PAKE (HPAKE), which enables the authentication server to detect password leakage. On top of the honeyword method, honey encryption, and OPAQUE, a standardised aPAKE, we also design an HPAKE structure. Our design's security is properly examined, with insider resistance and password breach detection achieved.

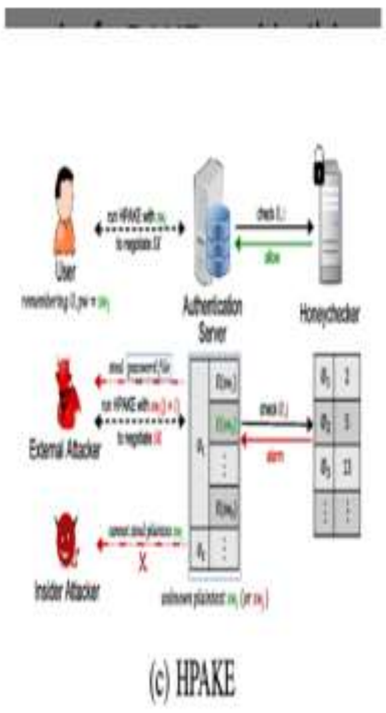


Fig 1:Architecture

3.1 IMPLEMENTATION

Online Shopping Website:

Using this module web application is developed which has online shopping features where seller can use admin module to upload products and buyer can view products and purchase. This application provides option for payment, add products to cart, view products, search products get conformation from admin on purchase, use attacker module to show internal atatcks. Show security methods to secure authentication process.

Admin Module:

This module is part on online shopping website where admin and login to application add products with cost and product details and verify users as attackers or normal users and block users who are attackers. Admin can verify users for purchasing products and get confirmation.

User Module:

This module is part of online shopping website where users can register with application by entering valid user name and password along with text file data which is used for every time login. User must give same text fiile every time and apply Honey Password-Authenticated Key Exchange when he login to application which will encrypt and send key to authentication server who will verity and validate user . If Password-Authenticated Key Exchange is success then only user is considered as normal user else he is considered as attacker.

Authentication Server Module:

This module is used as middle layer between user registration process and login process verification for verifying Honey Password-Authenticated Key Exchange process. Every time new user registers this server will store a security key which is unique based on user input data . If same data is uploaded by user while login then only authentication server will give validation else authentication exchange will be failed.

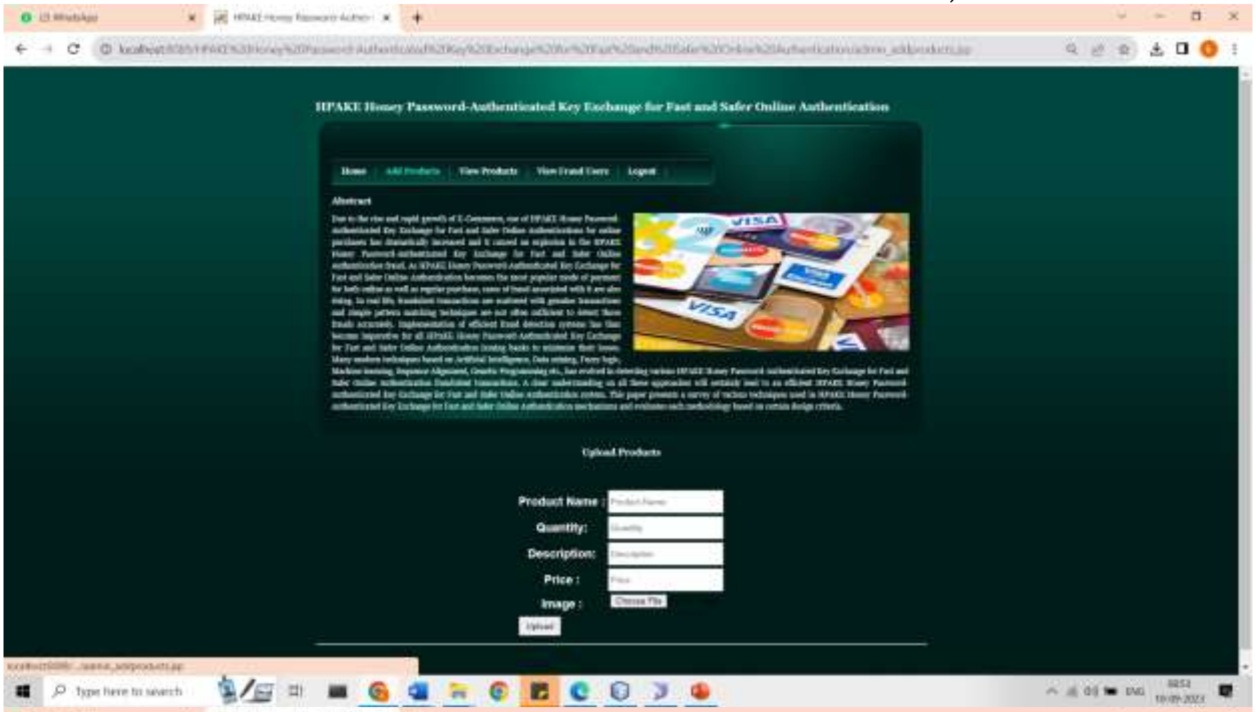
Honey key Mechanism:

Honeyword Mechanism Honeyword technique [15] has been proposed to detect password leakage for password-over-TLS. As shown in Figure 1b, honeyword mechanism directly generates several honeywords and stores them (in the form of hash value) on the authentication server along with the real password. It stores the index of the real password in the list on another server called honeychecker. When one logs in with a username U and a password pw (where pw is sent to the authentication server via the TLS channel), then the authentication server checks if pw is a sweetword: 1) If it is not, deny this login. 2) If it is the i-th sweetword, the authentication server sends (U, i) to the honeychecker (via a secure channel). Then the honeychecker checks if the index i is correct for U: a) If it is, allow this login. b) Otherwise, raise an alarm of password leakage and take actions according to the pre-defined security policy. This mechanism only does slight modification on the server side for password-over-TLS, and therefore maintains its advantages on deployability. Besides, since its interface is very simple, the honeychecker can be easily enhanced to avoid being compromised

Honey Key encryption:

Honey Encryption Honey encryption [25], [32] is a novel encryption method, which can yield decoy messages for incorrect keys as shown in Figure 4b. It introduces a probabilistic encoder to encode the message M to a (fixed-length) uniform bit string S and then encrypts S by a carefully-chosen traditional encryption scheme (see Figure 5). The encoder is designed according to the message distribution M, which can be uniform or nonuniform (e.g., for the password vaults [33]). The encoder should guarantee that decoding a random bit string will yield a message sampled from M. Formally, for an arbitrary adversary (maybe with unlimited computing resources) A, (M0, S0) and (M1, S1) are indistinguishable (we denote (M0, S0) ∼ (M1, S1)), where S0 ←$ {0, 1} l (i.e., randomly selecting a l-bit string), M0 ← Decode(S0), M1 ←p M (i.e., sampling a message from M according to the message distribution p), S1 ← Encode(M1), and l is the length of the bit strings. More specifically, |Pr[A(M0, S0) = 1 : S0 ←$ {0, 1} l , M0 ← Decode(S0)] − Pr[A(M1, S1) = 1 : M1 ←$ M, S1 ← Encode(M1)]| is negligible. Please note that the traditional encryption scheme used in honey encryption should yield a random bit string for each incorrect keys. Therefore, for each incorrect key K′ , the honey encryption scheme will produce a l-bit string S ′ and further a plausible-looking message M′ on M. In the design of HPAKE, we use honey encryption to encrypt the user's private key kU , which is a uniformly random number on Zm2 . Designing an encoder for kU is simple. To encode kU , we directly select an integer number from [round(kU 2 l/m2),round((kU + 1)2l/m2)) (⊆ [0, 2 l)) as S, where round is the rounding function; to decode S, we find the corresponding interval and obtain kU . With the encoder, for each incorrect key rw, the honey encryption scheme can produce a plausible-looking private key on Zm2

4.RESULTS AND DISCUSSION



HPAKE Honey Password-Authenticated Key Exchange for Fast and Safer Online Authentication

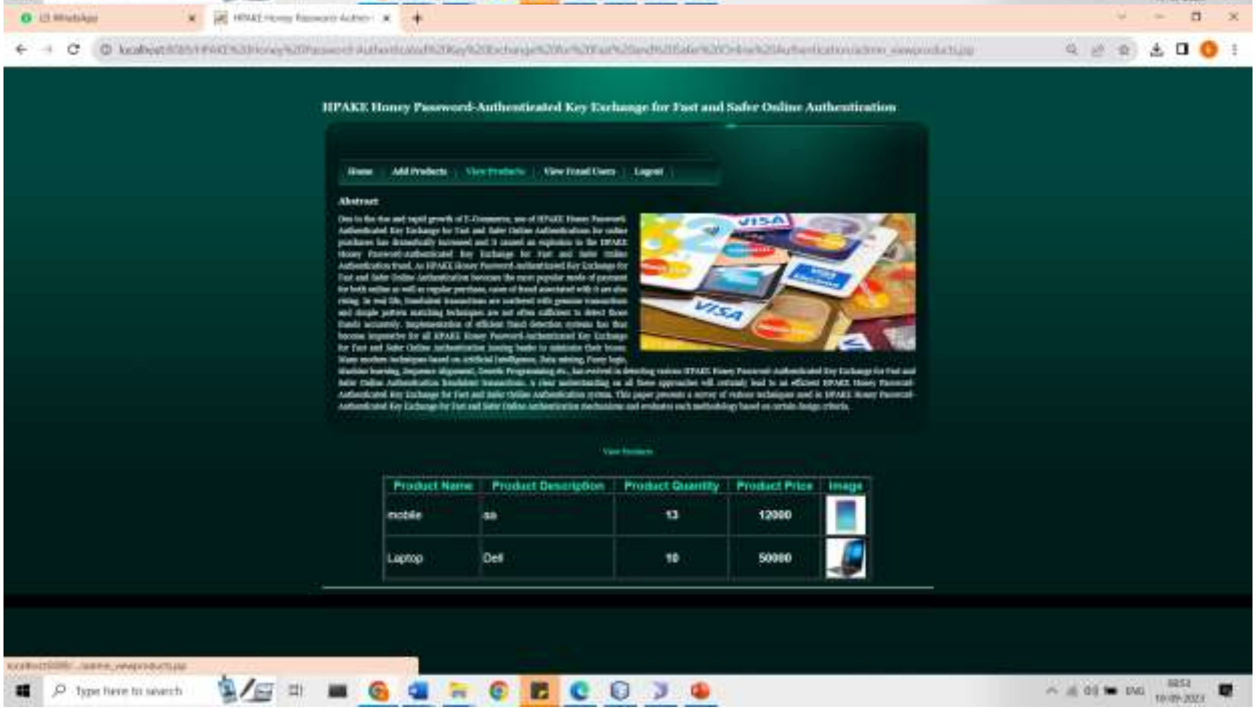
Home | Add Products | View Products | View Fraud Users | Logout

Abstract

Due to the rise and rapid growth of E-Commerce, use of HPAKE Honey Password-Authenticated Key Exchange for Fast and Safer Online Authentications for online purchases has dramatically increased and it caused an explosion in the HPAKE Honey Password-authenticated Key Exchange for Fast and Safer Online Authentication fraud. As HPAKE Honey Password-Authenticated Key Exchange for Fast and Safer Online Authentication becomes the most popular mode of payment for both online as well as regular purchase, cases of fraud associated with it are also rising. In real life, fraudulent transactions are scattered with genuine transactions and simple pattern matching techniques are not often sufficient to detect those frauds accurately. Implementation of efficient fraud detection systems has thus become imperative for all HPAKE Honey Password-Authenticated Key Exchange for Fast and Safer Online Authentication issuing banks to minimize their losses. Many modern techniques based on Artificial Intelligence, Data mining, Fuzzy logic, Machine learning, Sequence Alignment, Genetic Programming etc., has evolved in detecting various HPAKE Honey Password-Authenticated Key Exchange for Fast and Safer Online Authentication fraudulent transactions. A clear understanding on all these approaches will certainly lead to an efficient HPAKE Honey Password-Authenticated Key Exchange for Fast and Safer Online Authentication system. This paper presents a survey of various techniques used in HPAKE Honey Password-Authenticated Key Exchange for Fast and Safer Online Authentication mechanisms and evaluates each methodology based on certain design criteria.

Upload Products

Product Name :	Product Name
Quantity:	Quantity
Description:	Description
Price :	Price
Image :	Choose File



HPAKE Honey Password-Authenticated Key Exchange for Fast and Safer Online Authentication

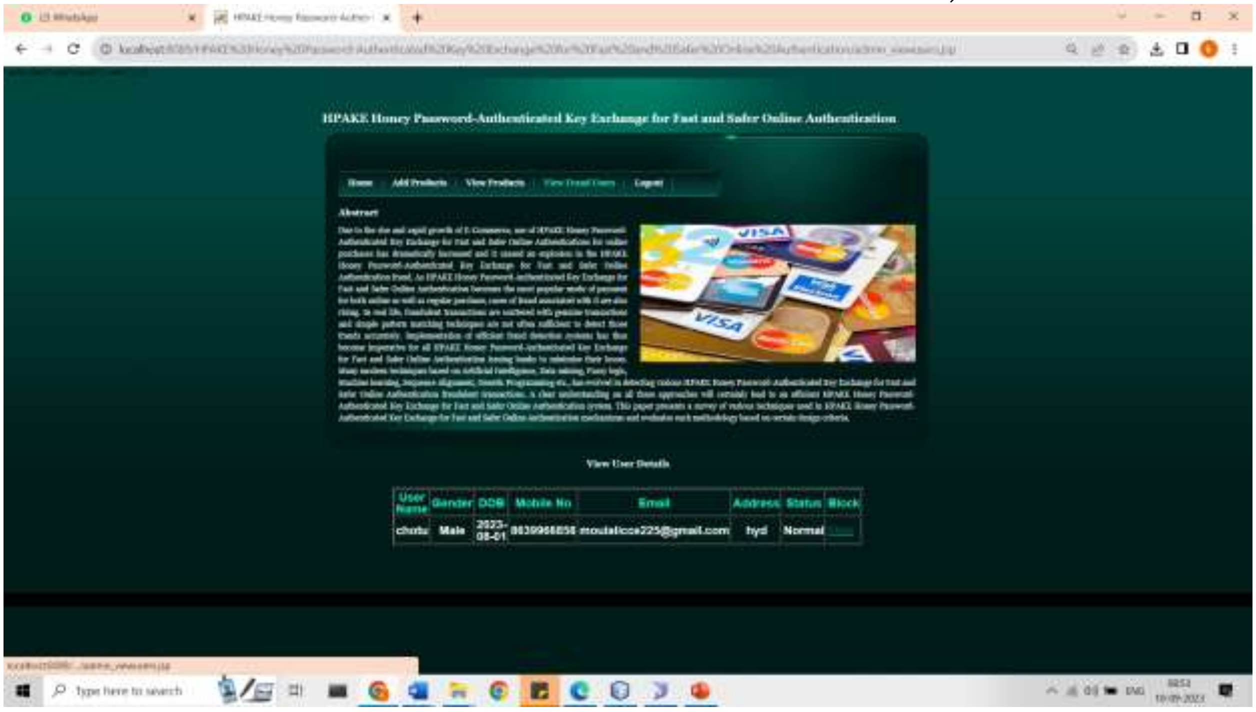
Home | Add Products | View Products | View Fraud Users | Logout

Abstract

Due to the rise and rapid growth of E-Commerce, use of HPAKE Honey Password-Authenticated Key Exchange for Fast and Safer Online Authentications for online purchases has dramatically increased and it caused an explosion in the HPAKE Honey Password-authenticated Key Exchange for Fast and Safer Online Authentication fraud. As HPAKE Honey Password-Authenticated Key Exchange for Fast and Safer Online Authentication becomes the most popular mode of payment for both online as well as regular purchase, cases of fraud associated with it are also rising. In real life, fraudulent transactions are scattered with genuine transactions and simple pattern matching techniques are not often sufficient to detect those frauds accurately. Implementation of efficient fraud detection systems has thus become imperative for all HPAKE Honey Password-Authenticated Key Exchange for Fast and Safer Online Authentication issuing banks to minimize their losses. Many modern techniques based on Artificial Intelligence, Data mining, Fuzzy logic, Machine learning, Sequence Alignment, Genetic Programming etc., has evolved in detecting various HPAKE Honey Password-Authenticated Key Exchange for Fast and Safer Online Authentication fraudulent transactions. A clear understanding on all these approaches will certainly lead to an efficient HPAKE Honey Password-Authenticated Key Exchange for Fast and Safer Online Authentication system. This paper presents a survey of various techniques used in HPAKE Honey Password-Authenticated Key Exchange for Fast and Safer Online Authentication mechanisms and evaluates each methodology based on certain design criteria.

View Products

Product Name	Product Description	Product Quantity	Product Price	Image
mobile	aa	13	12000	
Laptop	Dell	10	50000	



HPAKE Honey Password-Authenticated Key Exchange for Fast and Safer Online Authentication

Abstract

[illegible]

View User Details

User Name	Gender	DOB	Mobile No	Email	Address	Status	Block
chotu	Male	2023-08-01	8639966858	moulalicce225@gmail.com	hyd	Normal	[illegible]

5.CONCLUSION

We introduce the first-of-its-kind concept, HPAKE, which combines the best features of aPAKE and honeyword techniques: it can identify password leaks produced by outside attackers and keeps insiders from obtaining the password in plaintext. A concrete HPAKE structure is constructed by us using OPAQUE, honeyword mechanism, and honey encryption. We explicitly demonstrate the security of our design in a game-based security model that we develop in order to analyse the security of our design. We put the suggested plan into practice and make it available in the actual world. The outcomes of the experiments demonstrate the effectiveness of our approach in practical applications.

REFERENCES

[1] J. Bonneau, C. Herley, P. C. Oorschot, and F. Stajano, “The quest to replace passwords: A framework for comparative evaluation of web authentication schemes,” in Proc. IEEE S&P 2012, pp. 553–567.

[2] N. Huaman, S. Amft, M. Oltrogge, Y. Acar, and S. Fahl, “They would do better if they worked together: The case of interaction problems between password managers and websites,” in Proc. IEEE S&P 2021, pp. 1367–1381.

[3] D. Pasquini, A. Gangwal, G. Ateniese, M. Bernaschi, and M. Conti, “Improving password guessing via representation learning,” in Proc. IEEE S&P 2021, pp. 265–282.

[4] W. Li and J. Zeng, “Leet usage and its effect on password security,” IEEE Trans. Inform. Foren. Secur., vol. 16, pp. 2130–2143, 2021.

[5] “Have i been pwned?” [Online]. Available: https://haveibeenpwned.com

[6] “Yahoo! data breaches.” [Online]. Available: https://en.wikipedia.org/wiki/Yahoo! data breaches

[7] “Yahoo tries to settle 3-billion-account data breach with $118 million payout.” [Online]. Available: https://arstechnica.com/tech-policy/2019/0 4/yahoo-tries-to-settle-3-billion-account-data-breach-with-118-million -payout/

[8] Z. Whittaker, “Github says bug exposed some plaintext passwords,” https://www.zdnet.com/article/github-says-bug-exposed-account-passwor ds/, 2018.

[9] S. M. Bellovin and M. Merritt, “Augmented encrypted key exchange: a password-based protocol secure against dictionary attacks and password file compromise,” in Proc. ACM CCS 1993, pp. 244–250.

[10] V. Boyko, P. MacKenzie, and S. Patel, “Provably secure passwordauthenticated key exchange using diffie-hellman,” in Proc. EUROCRYPT 2000. Springer, pp. 156–171.

[11] C. Gentry, P. MacKenzie, and Z. Ramzan, “A method for making password-based key

exchange resilient to server compromise," in Proc. CRYPTO 2006. Springer, pp. 142–159.

[12] S. Jarecki, H. Krawczyk, and J. Xu, "OPAQUE: An asymmetric PAKE protocol secure against pre-computation attacks," in Proc. EUROCRYPT 2018. Springer, pp. 456–486.

[13] M. Abdalla, M. Barbosa, T. Bradley, S. Jarecki, J. Katz, and J. Xu, "Universally composable relaxed password authenticated key exchange," in Proc. CRYPTO 2020. Springer, pp. 278–307.

[14] S. Smyshlyaev, N. Sullivan, and A. Melnikov, "[cfrg] results of the PAKE selection process," 2020. [Online]. Available: https://mailarchiv e.ietf.org/arch/msg/cfrg/LKbwodpa5yXo6Vu NDU66vt Aca8/

[15] A. Juels and R. L. Rivest, "Honeywords: Making password-cracking detectable," in Proc. ACM CCS 2013, pp. 145–160.

[16] D. Wang, H. Cheng, P. Wang, J. Yan, and X. Huang, "A security analysis of honeywords," in Proc. NDSS 2018, pp. 1–18.

[17] Akshima, D. Chang, A. Goel, S. Mishra, and S. K. Sanadhya, "Generation of secure and reliable honeywords, preventing false detection," IEEE Trans. Depend. Secur. Comput., vol. 16, no. 5, pp. 757–769, 2019.

[1] .

.

Author's Profiles

#1:-Mrs.K.Harika working as Associative Professor in Department of IT in DVR & Dr.HS MIC College of Technology,Kanchikacherla-521180

#2:-S.Aparna(20H71A1204) B. Tech with Specialization of **Information Technology in DVR & Dr. HS MIC College of Technology, Kanchikacherla-521180**

#3:-A.Udaya Bhanu(20H71A1254) B. Tech with Specialization of Information Technology in DVR & Dr.HS MIC College of Technology,Kanchikacherla-521180

#4:-A.Harini(20H71A1209) B. Tech with Specialization of Information Technology in DVR & Dr. HS MIC College of Technology, Kanchikacherla-521180

#5:-M.Anil Kumar(20H71A1202) B. Tech with Specialization of **Information Technology in DVR & Dr. HS MIC College of Technology, Kanchikacherla-521180**