International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 03 | March - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Enhanced System for Securing Password Manager using Honey Encryption

ASWATHI V, ABHISHEK MANAS V, HARIGOVIND PK, JISHNU JITHESH, SIDHARTH APV
Department of Computer Science and (Cyber Security),
Vimal Jyothi Engineering College, Chemperi, Kannur

***Abstract*—**Honey Encryption (HE) has emerged as a cutting-edge cryptographic technique to thwart bruteforce assaults, by creating a believable fake decryptions for wrong keys and because of their limited entropy, traditional encryption methods like hashing and passwordbased encryption (PBE) frequently fall short against attacks on weak passwords. HE leverages DistributionTransforming Encoders (DTEs) to map plaintexts to seeds, ensuring secure encryption despite low-entropy passwords. Various advancements have been made to enhance HE, including expanding its message space, integrating with other cryptographic techniques such as AES, and optimizing it for password managers and natural language messages. Additionally, researchers have explored methods to mitigate HE's limitations, such as typo safety issues and computational overhead. The integration of HE with other security measures, including salt hashing, randomized keypads, and machine learning for intrusion detection, signifies its evolving role in cybersecurity. Recent studies also explore HE's application in protecting biometric data, securing cloud storage, and enhancing multi-factor authentication mechanisms. The findings suggest that while HE is a promising encryption technique, further refinements are necessary to enhance efficiency, usability, and resilience against evolving threats.

I. INTRODUCTION

As cyberattacks continue to rise in frequency and sophistication, securing digital infrastructure has become a critical challenge. Traditional encryption methods, while mathematically robust, often fail in practice due to human factors-particularly in password-based systems. Users frequently choose weak, predictable passwords that attackers can exploit using brute-force or dictionary attacks. This vulnerability highlights the pressing need for cryptographic solutions that can protect sensitive data even when passwords lack sufficient entropy.

To address this issue, Ari Juels and Thomas Ristenpart introduced Honey Encryption (HE), an innovative cryptographic approach that combines encryption with deception. Unlike conventional encryption, which either decrypts correctly with the right key or fails outright with an incorrect key, HE generates realistic but incorrect outputs-called honeywords-whenever an incorrect decryption key is used. These honeywords mimic the structure and format of the real data, misleading attackers and making it virtually impossible to distinguish between genuine and false decryption attempts.

The Distribution Transforming Encoder (DTE), a technique that converts plaintext communications to encoded seeds and makes sure that decryption attempts, whether successful or unsuccessful, produce believable results, is the central component of HE. This unique feature provides an additional layer of security by ensuring that even weak passwords result in seemingly valid data, significantly raising the bar for attackers attempting brute-force decryption.

1) Advancements and Research Directions: Honey Encryption has garnered significant attention in cryptographic research and security applications. Recent advancements focus on:

1) Enhancing DTE Algorithms: Improving the efficiency and scalability of DTE to support larger and more complex data distributions.
2) Integration with Established Cryptographic Systems: Combining HE with traditional encryption standards such as AES, biometric authentication, and password management systems to enhance overall security.
3) Usability Improvements: Addressing practical concerns such as typo tolerance, user experience, and computational efficiency to make HE more accessible for real-world applications.

4) Expanding Real-World Use Cases: Exploring applications in IoT security, cloud storage encryption, secure messaging, and protection of sensitive medical and financial data.

2) The Future of Honey Encryption: By incorporating deception as a core security mechanism, Honey Encryption offers a transformative approach to modern cryptography. As research progresses, further refinements in DTE design, usability, and system integration will determine HE's effectiveness in widespread adoption. By synthesizing theoretical advancements with practical applications, HE has the potential to reshape password-based encryption and provide robust protection against brute-force and dictionary attacks.

This review aims to provide a comprehensive analysis of Honey Encryption, shedding light on its principles, challenges, and future directions in securing digital communication and data protection.

II. LITERATURE SURVEY

Ari Juels and Thomas Ristenpart introduce [1] Honey Encryption (HE) as a novel cryptographic technique to address vulnerabilities associated with weak passwords, a recurring issue demonstrated by incidents like the RockYou breach Conventional techniques like hashing and password-based encryption (PBE) are vulnerable to offline brute-force attacks, especially when the entropy of the passwords is insufficient. HE mitigates this by creating ciphertexts that, when decrypted with incorrect keys, yield plausible but false outputs, preventing attackers from determining decryption success. Central to HE is the use of distributiontransforming encoders (DTEs), which map plaintexts to seeds, enabling secure encryption even with lowentropy keys. While the method has promising applications in securing password managers and sensitive data, it also faces challenges, including constructing robust DTEs for complex datasets and addressing typo-safety issues where incorrect passwords produce realistic but incorrect results. Recent research explores optimizing HE by incorporating machine learning techniques to improve the generation of honeywords and enhancing the scalability of DTEs. Additionally, studies suggest integrating HE with multi-factor authentication to further reduce the risks of password compromise.

The paper [2] introduces an enhanced Honey Encryption (HE) algorithm designed to address the message space limitations inherent in traditional HE methods, particularly in the Distribution Transforming Encoder (DTE) process. While the standard HE algorithm uses a cumulative distribution function (CDF) to map plaintext messages to a seed space, its effectiveness is limited when handling more than four messages due to constraints in satisfying probability distributions.The authors suggest employing a discrete distribution function in the DTE process to get around this, greatly increasing the message space and allowing HE to support a greater variety of applications.

The enhanced method also improves the honeywords generation algorithm to reduce storage overhead and resolve the typo safety problem in password-based encryption (PBE). The proposed approach minimizes the need for excessive storage by leveraging indexbased storage for real passwords and honeywords, reducing resource consumption. Additionally, by producing convincing fake messages for unsuccessful decryption efforts, the improved HE algorithm shows resilience against brute-force assaults, misleading attackers and boosting security. Through case studies and mathematical modeling, the paper validates the effectiveness of these enhancements, establishing a more scalable and secure framework for HE implementation in protecting sensitive data from bruteforce attacks. Additional experiments highlight improvements in computational efficiency and the ability to dynamically adjust probability distributions to fit various security contexts.

The paper [3] explores advancements in Honey Encryption (HE), emphasizing its role in securing sensitive data against brute-force attacks and other cyber threats. Initially developed by Ari Juels and Thomas Ristenpart, HE provides a defense mechanism that misleads attackers by generating plausible yet false decryption outputs (honeywords) for incorrect password attempts. This ensures that attackers cannot easily distinguish genuine keys from decoy ones, effectively burying the true password in a pool of fake possibilities.

The proposed advancements focus on making HE more effective by introducing special honeywords tailored to the attacker's decryption patterns. These enhancements include tracing the intruder's IP address and notifying the

victim and other potentially atrisk users about the security breach. By incorporating detection mechanisms alongside deflection strategies, the new system transitions HE from a purely defensive approach to an active deterrent, highlighting future risks to users and prompting them to strengthen their security measures.

The enhanced HE algorithm also addresses the limitations of traditional encryption systems in password managers and credit card data protection. By simulating user behavior and dynamically generating fake keys based on intrusion attempts, the system not only confuses attackers but also improves user awareness and response to cyber threats. This novel approach enhances the overall efficacy of HE, making it a robust and adaptable solution for modern cybersecurity challenges. Additional studies suggest applying behavioral analysis techniques to refine honeyword generation further, ensuring that responses remain contextually relevant to attacker behavior.

The integration of Honey Encryption (HE) with Advanced Encryption Standard (AES) [4] offers a robust solution to contemporary data protection challenges, especially in cloud environments. Honey Encryption introduces a mechanism to confuse intruders by providing plausible fake data, or ”honeywords,” when incorrect decryption keys are used, enhancing security against brute force attacks. Advanced Encryption Standard, though widely used for its symmetric encryption capabilities, is vulnerable to brute force and denial-ofservice (DoS) attacks due to its single-key dependency. By combining HE and AES, the system leverages honeywords as AES keys, creating multiple layers of security and significantly improving resistance to attacks.

Performance tests demonstrate the hybrid method’s effectiveness compared to standalone AES, though challenges such as computational overhead and memory requirements for Distribution Transforming Encoder (DTE) operations remain. Future research can focus on optimizing HE with lightweight algorithms to enhance usability in resource-constrained environments, including IoT devices, ensuring a balance between performance and security. Recent developments have explored ways to minimize HE’s storage and processing demands through probabilistic key distribution techniques, making it more practical for real-time encryption applications.

Honey Encryption (HE), introduced by Juels and Ristenpart in 2014 , [5] is a novel cryptographic scheme designed to protect data from brute force attacks by returning plausible fake messages, known as honeywords, when incorrect decryption keys are used. The approach leverages Distribution Transforming Encoders (DTE), which map plaintext messages to random seeds in a probabilistic manner, ensuring that attackers cannot distinguish between genuine and fake data.

This paper explores methods to enhance the applicability of HE, particularly in encrypting natural language messages, by proposing an API to simplify implementation and reduce the complexity of message space. A key innovation is the use of natural language processing techniques, including tokenization and word embeddings, to generate realistic fake messages, thereby improving the usability and security of HE. This enhancement reduces memory requirements, optimizes message distribution, and ensures fake messages are indistinguishable from real ones, offering an extra layer of protection even against sophisticated adversaries. By streamlining HE for practical applications, this study aims to encourage its adoption in modern cryptographic systems and expand its utility across diverse domains. Future improvements include exploring adversarial training techniques to refine honeyword generation based on real-world attack simulations.

[6] In an age where cyber threats are escalating, robust password protection has become critical for safeguarding personal and organizational data. The research highlights that 81 percent of data breaches stem from poor password security, necessitating innovative approaches to create and store secure passwords. Traditional human-generated passwords often follow predictable patterns, making them vulnerable to brute force attacks and other hacking techniques.

To address these concerns, this study explores the integration of salt and hashing techniques for password encryption, where random strings are appended to passwords before hashing, rendering them highly resistant to decryption attempts. It also emphasizes the need for randomized password generators, which produce complex alphanumeric and symbolic combinations, significantly reducing the risk of credential theft.

Furthermore, advancements like biometric encryption and systems such as oPass and SecureWeb are investigated, showcasing alternative authentication mechanisms that enhance security. The study underscores the importance of user awareness and education in adopting best practices for password creation and periodic updates to mitigate vulnerabilities in an increasingly digitized world. Additionally, researchers propose the incorporation of blockchain-based authentication systems to provide immutable password verification, further enhancing security.

The proposed framework [7] introduces a secure password mechanism using a reformation-based Honey Encryption approach combined with a dynamic keypad system to enhance data security. The dynamic keypad rearranges characters for each login, making passwords unpredictable and resistant to brute force, shoulder-surfing, and other attacks. This mechanism ensures that even if an attacker intercepts a password, it remains useless due to its constantly changing nature. Honey Encryption ensures that incorrect decryption attempts yield misleading outputs, enhancing robustness by preventing adversaries from knowing whether they are close to the correct password. Passwords are generated with complex alphanumeric patterns and reformed into encrypted versions for added security, providing additional layers of protection beyond standard encryption methods. While the framework faces challenges like computational overhead and increased verification latency, it outperforms traditional methods by providing superior resistance to advanced attacks. Furthermore, the inclusion of a dynamic keypad minimizes the risk of visual hacking and shoulder surfing. Compared to existing techniques, such as secure password storage and quantum-resistant protocols, this system offers an innovative approach to dynamic security. Future improvements could include integration of machine learning for unauthorized access detection, personalized password suggestions based on user behavior, and scalability for enterprise-level applications, offering a more resilient and adaptable security solution.

The paper [8] titled ”Password Typos Resilience in Honey Encryption” by Hoyul Choi et al. addresses the limitations of Honey Encryption (HE) in handling password typos. HE, a novel encryption scheme proposed by Juels and Ristenpart, secures data against brute-force attacks by producing fake but plausible messages upon incorrect password attempts. However, this characteristic complicates scenarios where legitimate users make typographical errors, as it becomes challenging to distinguish between valid and erroneous decryptions. The authors suggest the A-Type and B-Type methods as two prototypes for identifying password typos. Although the A-Type approach has a smaller key size and less typo detection uncertainty, it is effective and appropriate for traditional client-server architectures. Conversely, the B-Type scheme, which incorporates additional user-side information such as a PIN, ensures higher typo detection accuracy but burdens users with additional memory requirements. Both schemes maintain message recovery security and offer adaptability for various applications. The paper underscores the importance of addressing the typo issue in HE to enhance usability while retaining its robust security features. It also explores potential solutions for balancing user convenience with system complexity, aiming to provide a smooth experience without compromising security. Future research may focus on improving typo detection accuracy without overwhelming the user and optimizing key management to prevent performance degradation.

Shannon’s foundational work [9] on information theory established the concept of perfect secrecy, achieved through uniform key distributions and message sets of equal cardinality, as exemplified by Vernam’s one-time pad encryption. However, practical systems often face challenges when dealing with nonuniform message distributions or limited key sizes. In order to solve this, Juels and Ristenpart developed Honey Encryption, which maps non-uniform messages to a wider set of seeds with a distribution that is almost uniform using a Distribution Transforming Encoder (DTE). This approach mitigates an adversary’s advantage by ensuring plausible decodings for every ciphertext, making it extremely difficult for an attacker to distinguish between real and fake messages. While Honey Encryption finds applications in securing passwords and biometric templates, it relies heavily on auxiliary randomness and requires a large number of seeds, making it resource-intensive. To overcome these limitations, the reviewed paper proposes a message partitioning technique that reduces adversarial success probabilities by dividing messages into smaller partitions.

This method is particularly effective in environments with limited randomness and can operate independently or in conjunction with DTE to balance security and resource efficiency. The partitioning strategy provides a more efficient use of available randomness while maintaining robust security levels. However, challenges remain, including the need for more efficient DTE schemes, minimizing randomness requirements, and establishing tighter bounds to determine theoretical performance limits. Future work aims to address these challenges and further optimize these encryption techniques, such as integrating AIdriven randomness management to dynamically adjust security measures according to the environment.

An enhanced architecture to counteract brute-force attacks on password managers is proposed in the paper [10] ”Enhanced System for Securing Password Manager Using Honey Encryption”.The system uses Honey Encryption (HE) in conjunction with other defenses to identify, stop, and deceive attackers. When an incorrect master password is entered, the system generates plausible honeywords, compares them with the entered password, and redirects intruders to a decoy account containing dummy data, ensuring that attackers cannot gain access to real user information. The framework explores three scenarios: legitimate access with correct credentials, detection and redirection to a decoy account upon honeyword matches, and account suspension after three incorrect attempts. The study compares the performance of HE combined with techniques like OTP, Blowfish, and homomorphic encryption. Results reveal that while combining HE with other security techniques slightly increases execution time, it significantly enhances confidentiality and integrity by providing a multi-layered defense. Challenges include distinguishing legitimate users with typographical errors from attackers and preventing attackers from recognizing the encryption system. Additionally, ensuring seamless user experience with minimal disruption due to security measures is a challenge. Future research will focus on using machine learning for dynamic decoy generation and improving detection accuracy by tracking intruder activity, such as IP addresses. Moreover, the system may be further enhanced with adaptive behavior analysis, using historical data to refine security decisions and improve threat detection in real-time.

TABLE I COMPARISON TABLE

Reference	Description	Advantages	Disadvantages
[1]	Introduced Honey Encryption as a method to protect weak passwords by generating plausible fake outputs for incorrect decryption attempts, using DistributionTransforming Encoders (DTEs).	• Provides robust defense against brute-force attacks. • Generates plausible decoy outputs, misleading attackers. • Useful for securing low-entropy data like passwords. • DTE ensures security even with weak keys.	• Difficulties in constructing robust DTEs for complex datasets • Typo-safety issues may inconvenience legitimate users. • Limited efficiency in certain large-scale applications.
[2]	Focuses on overcoming HE’s limitations by using discrete distribution functions in DTEs, improving storage efficiency, typo safety, and scalability for broader applications.	• Expands message space using discrete distribution functions in DTE. • Minimizes storage overhead by utilizing	• Increased complexity in DTE implementation. • Computational overhead may affect real-time applications.

		efficient index-based storage. • Resolves typo-safety issues with improved honeyword generation. • Strengthens resistance to bruteforce attacks.	
[3]	Proposes attacker-tailored honeywords, intruder detection mechanisms, and dynamic decoy strategies, transitioning HE into an active deterrent system.	• Adapts honeywords to attacker decryption patterns. • Enhances security by combining detection mechanisms and decoys. • Improves user awareness and strengthens security measures.	• Detection mechanisms may introduce latency. • Dependence on accurate intrusion patterns for effectiveness.
[4]	Combines Honey Encryption and Advanced Encryption Standard (AES) to enhance cloud data protection, addressing brute force and denial-of-service vulnerabilities.	• Combines HE's decoy strategy with AES's symmetric encryption. • Provides layered security against brute-force and DoS attacks. • Demonstrates superior performance compared to standalone AES	• Higher computational and memory requirements for DTE operations. • Overhead may limit usability in IoT and resource-constrained environments.
[5]	Explores using natural language processing to simplify HE implementation, generating realistic fake messages to secure sensitive information.	• Enhances usability and security for encrypting natural language messages. • Reduces memory usage with optimized message distribution. • Improves fake message realism using NLP techniques.	• Implementing NLP-based DTE involves significant complexity. • Potential challenges in ensuring perfect message indistinguishability..
[6]	Discusses salt and hashing for securing passwords, alongside innovative authentication mechanisms like biometrics and randomized password generation.	• Salt and hashing techniques enhance resistance to decryption attempts. • Randomized password generators improve password complexity.	• Does not directly integrate with HE. • Relies heavily on user compliance and awareness.

		• Presents alternative methods, including biometric encryption.	
[7]	Introduces a dynamic keypad with Honey Encryption to thwart brute force and shoulder-surfing attacks, emphasizing adaptability and resistance to advanced threats.	• Dynamic keypads prevent bruteforce and shoulder-surfing attacks. • Combines alphanumeric password generation with HE for robustness. • Outperforms traditional methods in advanced attack scenarios.	• Increased verification latency and computational overhead. • Usability challenges for frequent logins.

TABLE II COMPARISON TABLE

Reference	Description	Advantages	Disadvantages
[8]	Addresses typo safety in Honey Encryption with AType and B-Type schemes to detect and manage typographical errors during decryption attempts.	• A-Type and B-Type schemes improve typo detection. • Enhances user experience by addressing typo-related inconveniences. • Maintains security against bruteforce attacks.	• A-Type scheme reduces key size and typo detection accuracy. • B-Type scheme increases memory requirements for users.
[9]	Highlights how HE leverages Shannon's principles and DTEs to map non-uniform messages securely, proposing message partitioning for enhanced efficiency.	• Mitigates adversarial success through message partitioning. • Minimizes the need for randomness in resource-constrained systems. • Balances security and efficiency with partitioned message spaces.	• DTE construction is resourceintensive. • Challenges in optimizing performance for large datasets.
[10]	Proposes a framework combining HE with decoy accounts and detection mechanisms, enhancing password manager security against brute force attacks.	• Redirects attackers to decoy accounts, enhancing confidentiality. • Combines HE with OTP and homomorphic	• Execution time increases with multi-layered techniques. • Typo detection remains a challenge in

		encryption for added security. • Tracks intruder activity to improve detection accuracy.	distinguishing attackers from users.

III. CONCLUSION

Honey Encryption has fundamentally changed how we approach data security by introducing a creative way to counter brute-force attacks. By generating plausible but incorrect decryption results through the use of Distribution Transforming Encoders (DTEs) and honeywords, HE safeguards sensitive data—even in cases where weak passwords are used. Over time, significant advancements have addressed HE's early limitations, such as restricted message spaces and vulnerabilities to password typos. Enhanced DTE designs, dynamic decoy generation using machine learning, and integration with encryption methods like AES have all expanded HE's utility and strengthened its defenses against increasingly sophisticated cyber threats.

HE's applications have broadened significantly, encompassing password managers, biometric data, natural language processing, and cloud-based security systems. Recent developments, such as typo-resilient protocols and adaptive frameworks like dynamic keypads and IP-based tracking, showcase HE's versatility as a robust security solution. However, challenges persist, particularly in minimizing computational demands, optimizing DTE construction, and improving usability.

Future studies should concentrate on improving HE's scalability for resource-constrained situations, such as the Internet of Things, and honing techniques to differentiate between attackers and authorized users. By overcoming these obstacles, HE can gain traction as a cyber security and data protection technology.

REFERENCES

[1] Juels, Ari, and Thomas Ristenpart. "Honey encryption: Encryption beyond the brute-force barrier." IEEE security and privacy 12.4 (2014): 59-62.

[2] Moe, Khin Su Myat, and Thanda Win. "Enhanced honey encryption algorithm for increasing message space against brute force attack." 2018 15th international conference on electrical engineering/electronics, computer, telecommunications and information technology (ECTI-CON). IEEE, 2018.

[3] Piyush. "Advanced Honey Encryption: An Escape-less Trap for Intruders." 2018 4th International Conference on Computing Communication and Automation (ICCCA). IEEE, 2018.

[4] Sarmila, K. B., and S. V. Manisekaran. "Honey encryption and AES based data protection against brute force attack." 2022 Sixth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC). IEEE, 2022.

[5] Nagraj, S., et al. "An approach towards making Honey Encryption easily available." 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN). IEEE, 2020.

[6] Chakraborty, Suryadip, et al. "A Study on Password Protection and Encryption in the era of Cyber Attacks." SoutheastCon 2024. IEEE, 2024.

[7] Nirmalraj, T., and J. Jebathangam. "A Password Secure Mechanism using Reformation-based Honey Encryption and Decryption." 2022 International Conference on Inventive Computation Technologies (ICICT). IEEE, 2022.

[8] Choi, Hoyul, Hyunjae Nam, and Junbeom Hur. "Password typos resilience in honey encryption." 2017 International conference on information networking (ICOIN). IEEE, 2017.

[9] Ghassami, AmirEmad, Daniel Cullina, and Negar Kiyavash. "Message partitioning and limited auxiliary randomness: Alternatives to Honey Encryption." 2016 IEEE International Symposium on Information Theory (ISIT). IEEE, 2016.

[10] AlMuhanna, Albatoul, Afnan AlFaadhel, and Anees Ara. "Enhanced system for securing password manager using honey encryption." 2022 Fifth International Conference of Women in Data Science at Prince Sultan University (WiDS PSU). IEEE, 2022.