1. Introduction

Despite having more security flaws than other authentication methods, password-based authentication has dominated authentication schemes for decades. Password-based authentication is still used by users because of its usability [1]. Users expect a scalable, scalable authentication system, easy to learn, needs less memory, and does not have anything to carry. However, the user password's security flaws continue to become a significant concern. Password-based authentication is familiar to the user but is vulnerable to password-guessing attacks.

The password-guessing attack is one of the attack scenarios when the attacker attempts to gain access to legitimate users' resources by guessing all possible passwords [2]. The attacker generates all the password combinations based on the password space, user-defined dictionary, or using users' personal information to find the correct password. To mitigate this attack, users have to choose a good password that is hard to guess. However, higher security usually means low usability. A password secured enough from password-guessing attacks will be difficult for users to memorize.

One of the prior works that increase the complexity of password-guessing attacks without decreasing the usability is honey encryption [3]. Honey Encryption (HE) produces a "honey message" when the attacker guessed the incorrect password. "Honey message" is a fake plausible-looking plaintext that makes the attacker believes that he inserted the correct password. This method makes the attacker couldn't confirm which one is the correct password from the guessed password list, such that the password-guessing attacks are hard to perform. By using this method, users still can create the password they desired without security concerns.

One authentication scheme that implements the HE is a secure pin authentication in java smart card by Mohammed [4]. This method prevented password-guessing attacks by limiting the number of login attempts and creating multiple honey data. The honey data is fake data with the same type of information stored in the smart card. If the limit of login is reached, then honey data is returned. This action makes the attacker believes that he successfully retrieve the correct data. However, this research does not mention how to create the fake data, and all of this fake data is stored in the smart card, such that the complexity of the security is limited to the number of fake data that can be stored.

An alternative approach that implements the HE is an authentication scheme proposed by Jordan [5]. This method prevented password-guessing attacks by generating sweet words from users' passwords and using the honey words. A honey word is generated and used as a fake message for each incorrect password. Suppose the attacker sends an incorrect password that is included in the sweet words. In this case, the corresponding fake message is returned to the attacker, and an alert is sent to the administrator. This method could give the attacker confidence that he successfully found the correct password. However, the fake message just using one word could raise suspicion of the attacker. All sweet words and decoy messages are also stored in the database, such that the security is limited to the number of fake messages that can be stored.

The security of implemented HE in the prior authentication system has to be increased to prevent password-guessing attacks. The honey message has to look natural enough to fool the attacker. The message must also be dynamically generated, so the choice is not limited to the stored fake messages. This improvement must be performed without decreasing the users' options in creating a password.

In this research, we proposed an authentication system using a natural language-based confirmation message. The proposed method changed the confirmation message into a honey sentence using natural language. Instead of rejecting the unattended request, the attacker received a honey sentence, such that he could not determine the correctness of the guessed password. The honey sentence is dynamically generated, such that one desired password from users could generate multiple types of sentences. One of these sentences is used as a seed sentence and stored in the database. The seed sentence is a sentence that is used to generate another possible sentence for the password. Generating multiple types of sentences is proposed to hide the pattern of sentence generation.

The experiment result showed that 80,67% of the generated sentences are natural. The complexity of password-guessing is increased by the number of possible honey sentences, such that the probability of finding the correct password from all possible passwords decreases.

The rest of the paper is organized as follows. Section 2 discusses the implementation details of previous methods, and Section 3 discusses the implementation details of the proposed method. Section 4 discusses the experimental results as well as the proposed method's security analysis. Finally, in Section 5, we present our work's research findings and conclusion.