Honey2Fish - An enhanced hybrid encryption method for password and messages

MSc Research Project
Cyber Security

Somil Jain
Student ID: X21154350

School of Computing
National College of Ireland

Supervisor: Dr. Rohit Verma
[Asst. Professor
School of Computing]

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Somil Jain
Student ID:	X21154350
Program:	MSc in Cyber Security
Year:	2022
Module:	MSc Research Project
Supervisor	Dr. Rohit Verma
Submission Due Date:	15th December 2022
Project Title:	Honey2Fish - An enhanced hybrid encryption method for password and messages
Word Count:	6899
Page Count	21

Signature: Somil Jain

Date: 15th December 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on the computer.	□

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Honey2Fish - An enhanced hybrid encryption method for password and messages

Somil Jain
X21154350

Abstract

Nowadays, there is an upsurge in cyber assaults due to increased Web usage. One of the most essential considerations is to secure password protection systems. Extensive research on strong passwords, secrecy, and reliability has been undertaken, yet several concerns remain unresolved. The principle of cryptography has already been broadly accepted, although it does carry certain risks, including the possibility of communication interception if somehow the key is compromised. The suggested model tries to create a combination of Honey encryption (HE) with Twofish. Besides this, HE is included to prevent brute force. Two-layer protection is presented in this research to protect the privacy of credentials as well as the message. As honeyword is utilized to give bogus but justified credentials or data when encrypted with an incorrect key, the security is improved. Typos are rather visible while entering the password, which may fool real users. This research also looks into contemporary honeyword creation algorithms as well as how HE typos might be avoided. A simple-to-implement honey word creation technique is also one of the main focuses of this research.

1 Introduction

Since there has been a rapid increase in internet and technological usage, the privacy, validity, and security of the credentials are crucial. Several security practices, such as input validation, session management, authentication, and authorization, should be considered. There are many methods to enhance security, Encryption is one of them. Data is turned into an unreadable form, known as ciphertext, using the notion of cryptography. There are many different types of cryptographic procedures. There are essentially two main forms of encryption: symmetric encryption and asymmetric key. Password-based encryption, often known as PBE, is commonly used by many organizations in the contemporary digital age to protect credentials and confidential information. The majority of user-generated passwords have low entropy. Users frequently tend to repeat their previous passwords and select quick-to-remember credentials. Poor credential entropy is indeed the usual outcome of this, making the PBE vulnerable to a Brute force assault. Attacks using brute force are quite dangerous and can result in the theft of confidential information.

One of the techniques made advantage of a statistical coding strategy to produce fake plain text using the ASCII (American Standard Code for Information Interchange) code table, but since the plain text was not encrypted, the attacker could readily determine which password was valid and which was incorrect. Users frequently repeat their previous passwords and select quick-to-remember credentials. Poor credential entropy is indeed the usual outcome of this, making the PBE vulnerable to a Brute - forced assault. Attacks using brute force are quite dangerous and can result in the theft of confidential information. One of the techniques made advantage of a statistical coding strategy to produce fake plain text using the ASCII (American Standard Code for Information Interchange) code table, but since the plain text was not encrypted, the attacker could readily determine which passwords were valid and which was wrong. The concept of using two encryption techniques called the hybrid model is prominently used nowadays. Figure 1 shows a general hybrid model in which two encryption technique is used.

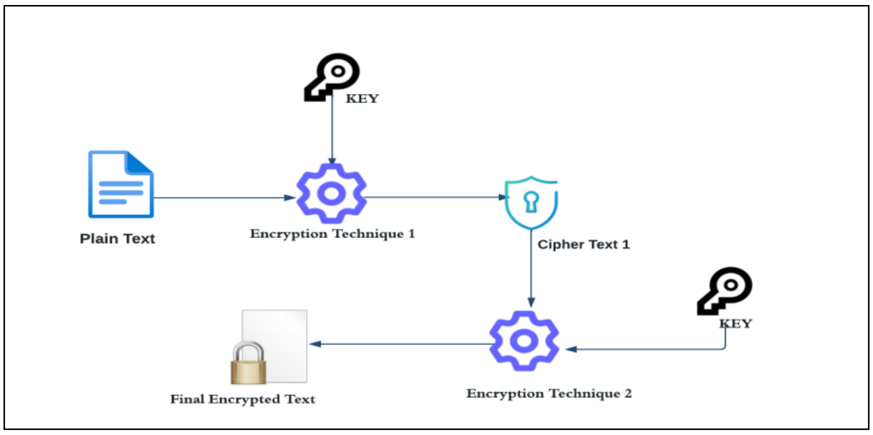


Figure 1: Generalized hybrid encryption model.

In the study (Erguler, 2016), the author creates Honey Encryption (HE), which will operate as a secondary protective layer. Existing systems are designed in such a way that provides security by keeping credentials in an encoded format, which serves as a single safeguarding layer. A honey encrypting mechanism is developed to offer protection against such an attack. It's indeed conceivable to think that honey encryption is like a strategy that will mislead an intruder regarding genuine and correct credentials. Whenever the entropy of the credential is very low, HE is beneficial. Honey encryption (HE) is however employed to safeguard the availability of systems because brute-force attempts by hackers frequently result in Attacks causing server unavailability.

However, because of the usage of honey words, which are bogus content put in the servers, the attacker will be fooled by receiving misinformation. The usage of HE is particularly advantageous in password-based encryption (PBE), however, the term "password" is frequently used vaguely in this context. PINs, credit card information, authenticator, and other low minimum entropy factors can all be utilized with HE. It might also be useful as a buffering to prevent keys having high entropies from being compromised entirely. A Distribution Transforming Encoder (DTE) was employed by the pre-existing models to enable Honey encryption. Additionally, the first and most recent approach integrates AES and honey encryption, which requires higher computation and calls for a variety of communication cycles. Furthermore, RSA pins, credentials, and even information about credit cards are encrypted using the honey system. HE must be altered to be utilized in a variety of situations, including bots for personal data like emails and alluring decoys to avoid eavesdropping during discussions. It is clear from (Dibas and Sabri, 2021), that Twofish is a better option when compared to AES because it possesses a special blend of elements like conservative structure, efficiency, quickness, and adaptability. The Two fish encryption method is thought of as very strong and particularly resistant to any key-related attack, including related-key various attacks and slide attacks since no keys can be reprocessed for any related assault. The researcher of some other work (Vivek Raj et al., 2020) utilizes the RC2 algorithm, however, the drawback of RC2 is that sometimes it requires a lot of cycles even though it produces results that are preferable to AES.

1.1 Motivation

Due to authentication flaws caused by the use of inadequate security practices, cyber attackers may be able to access a system without permission. Passwords are susceptible to a variety of attacks even with strong security measures like complex hashing techniques and strong data encryption. The main motive of this research is to develop a hybrid combination that secures both saved messages and passwords. The message's security is improved by the use of the honey encryption technique, which guards against any kind of brute force assault. The primary reason for adopting honey encryption is that it is still a relatively untested technique with little research being done. A novel hybrid model of two fish encryption and honey encryption techniques will be presented in the study.

1.2 Research Question

If combined with Two Fish encryption, how may Honey encryption aid in enhancing the protection of passwords and messages?

1.3 Structure of the Report

In the following section 2, a comprehensive research study (literature review) addressing honey encryption is shown. The related work emphasizes the shortcomings and the way they were gradually rectified. The research is carried out by taking into consideration DTE, hybrid models that already exist, honeyword production strategies, and a detailed discussion of the constraints of these models.
The next section 3 discusses the research methodology related to simple honeyword creation techniques and how a very simple-to-implement DTE could be constructed. After this, the implementation is discussed in which the snippets of the output including the Twofish and honeyword creation and message retrieval processes are discussed.
After implementation, the evaluation of the proposed model is done. The evaluation is done by calculating the variation in encryption, and decryption time if varying the password size. The performance of AES and the Twofish algorithm are also discussed. The paper ends with a conclusion and learning of the proposed model along with the references.

2 Related Work

Nowadays, password attacks are increasing at a rapid pace. It is observed that over 3 billion passwords are exposed every second. There are a lot of encryptions and hashing techniques which are used for securing both the data at rest and data in motion. To secure password storage one of the techniques that can be used is honey encryption which makes use of honeywords also called sweet words. This is also known as the concept of deceiving the attacker. Considerable use is made of hybrid models, which integrate two or more methodologies. A thorough overview of Honey encryption, issues with its implementation, and the Twofish encryption will be covered in the parts that follow.

2.1 Honey Encryption

Fake information is frequently referred to as "honey" in information security. A honeypot is the most popular illustration of honey; it is a dummy server intended to entice intruders, divert their attention away from actual targets, and perhaps even detect them. Credential security has been recommended using honeywords also called sweet words. For every user, a number of fake passwords would've been kept in addition to the real one, or just the "sweet words. Honey encryption was used by (Juels and Rivest, 2013) for the very first time. They make use of the fake passwords stored with the real passwords which eventually created the problem of storage overhead. Figure 2 below shows the mapping of seed (00,01,10,11) with the messages(RED, BLUE, GREEN, WHITE).

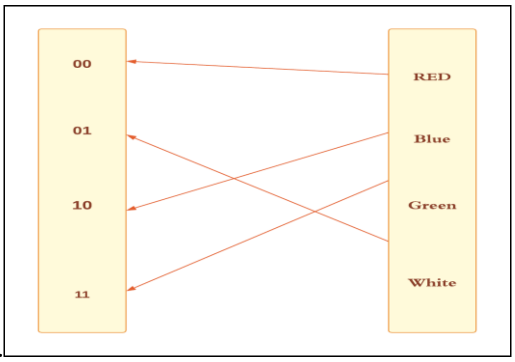


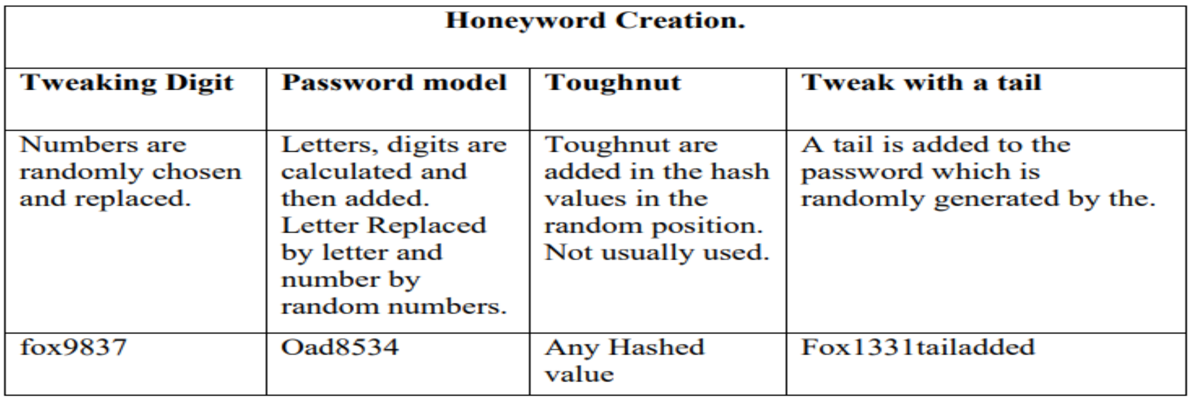
Figure 1: Message Encoding in Honeyword Encryption

It also impacted the computational time and overall data retrieval. Other solutions are proposed to make use of servers for password storage. (Chakraborty and Mondal, 2017) the proposed similar model called Honey circular which makes use of the Honey Circular List. This method makes use of the distance of password storage over a circular list and somehow if the hacker finds the honey words list, then he will never find the real password as the distance is still unknown. The problem of storage was addressed but the honeyword generation algorithm still needed to be addressed.

2.2 Honeyword Creation Techniques

Now, honey encryption was used but still, but the creation techniques of honeyword are still unexplored. (Pagar and Pise, 2017), these researchers discussed the honeyword creation methods which include Chaffing passwords with tough nuts, tweaking, etc. Many such models are used after that, some used a hybrid model like in the paper, (Erguler, 2016) makes use of chaffing with a password in which the sweet words are created by changing, or manipulating the password by removing, adding, and altering some letters. It makes use of the original password. This technique has shown some improvement but still, it is not as effective and accurate since it makes use of honeyword, and passwords are directly stored in the same files.

Table (1) below shows the different creation techniques (Erguler, 2016).



Honeyword Creation.			
Tweaking Digit	**Password model**	**Toughnut**	**Tweak with a tail**
Numbers are randomly chosen and replaced.	Letters, digits are calculated and then added. Letter Replaced by letter and number by random numbers.	Toughnut are added in the hash values in the random position. Not usually used.	A tail is added to the password which is randomly generated by the.
fox9837	Oad8534	Any Hashed value	Fox1331tailadded

Table 1: A table shows honeyword creation techniques

Another issue in honey word implementation was typos also called typological errors. These errors are one of the major problems that needed to be considered. This is mostly due to the user’s typing mistake, and this may lead to confusion between genuine and fake users. (Chen et al., 2016) provides a solution to typos by which the author used two types of protocols to fix these typos. Although the problem was solved still the algorithm used is not that effective when it comes to performance and reliability.

2.3 Distribution Transforming Encoder (DTE) with its shortcomings

Many researchers use the concept of DTE which implements a function that is discrete and combined with another security concept. In a research project by (Shen et al., 2016), a percentage of participants tend to utilize their earlier login details throughout all different profiles. Since the method used earlier generates a fresh honeyword each time, this practice results in the ambiguity between different systems that share the very same passcodes. In HBAT (Honeyword-based authorization technique), this is referred to be multiple system vulnerabilities. (Chakraborty and Mondal, 2017) , developed a novel honeyword-generating method that he named "Paired Distance Protocol" (PDP). This writer's core thesis is to reduce the overhead storage load required to produce such honeywords. PDP considered the space issue, parameters for code reuse, and robust and reliable design. One minor disadvantage is that extra information needed to be provided by the user to do password tailing.
One such solution was proposed by (AlMuhanna et al., 2022), in which the password is checked with respective honeywords and if it gets matched with any honeywords then the user is flagged, and the alert is sent to the admin. One problem with this method was that if a real user mistakenly typed the wrong password, then the intruder will flag the user.
Another model by (Mohammed et al., 2020), make use of java applets to create java card using honey encryption authentication techniques. The code is designed in such a way that a threshold is selected and if any attacker reaches that threshold, he will be given any random page like the original page. The author of the paper provides a mitigation against the brute force attack which eventually helps to protect the availability of the data. It is also advised in this paper to use a machine learning algorithm instead of using fake data stored in servers.

(“A Novel Hybrid Encryption Method Based on Honey Encryption and Advanced DNA Encoding Scheme in Key Generation,” Nwe Ni Khin, Thanda Win.), presents proof of DNA encoding. The proposed approach is a hybrid model that has honey encoding with a DNA cryptosystem. As an outcome, the issue with the associated data limitations might be resolved, and the suggested method can withstand attacks like brute force. Additionally, the modernized DNA coding strategy to the honey encrypting system's public key. In order to provide security that goes above attacks by brute force and also to execute code on encrypted files, respectively proposed system by (Bangera et al., 2020), uses both honey encryption and homomorphic encryption keys. These methods contribute to far greater data secrecy and substantially more reliable authenticity.

The dynamic keypad scheme is used by, (Nirmalraj and Jebathangam, 2022) to present difficulties for attackers. Given the dynamic keypad, it also took a substantial amount of time to extract the passwords. As a result, attackers won't be able to rapidly figure out the credentials utilizing the suggested approach, which will use algorithms for machine learning for future inscriptions and descriptors.

To defend the server from man-in-the-middle cyberattacks as well as brute force attacks (Tan et al., 2021), implemented honey encryption, grid-based passwords, and OTP techniques. To enhance mitigation even further over surfing and to respond to attacks, the author presented a technique that included grid-based HE. Along with suggested Grid-based Honey Encryption, a fuzzy false health record that perfectly matches that of actual patients is produced rather than preventing an attacker from viewing the patient's actual report. (AlMuhanna et al., 2022) suggested a method in which credentials are kept utilizing hashes, and provided passcode satisfied the need, this will lead to generating a specific sweet word and saving them for future use. Thereafter, multi-layer authentication is carried out when any user wants to sign up for a new account. A security question will appear when you enter your password, and your response will serve as the 2nd layer of authentication. After several unsuccessful attempts to log in with an invalid password, a bogus deceptive webpage that imitates the real is displayed. The "semantic security-style notion," which is particularly helpful for unknown threats, and then another assumption, such as the target distribution, which is not changeable, are both used in the new proposal by(Jaeger et al., 2016).
(Chen et al., 2016), suggested an innovative model in which a MIMS (Mobile Instant Messaging Service) to communicate and exchange information. The researchers designed MIMS using 1 Time Pad cryptography algorithms. RTC-MTT and other Crypt21 techniques are also used by him. For the generation of OTP, the author utilized a password with encryption. The study recommended using the EL-GAMAL protocol to generate session keys in order to increase interaction effectiveness and safety.

The fake Data Generation Algorithm is developed by(Sahu, 2020) in order to protect the system from brute force attacks. In this, the author created a fake message in order to protect the information from intruders. (Omolara et al., 2018) NLTK from Stanford and Wordnet from Princeton is used by the author in which the HE is used to decode the emails. The generation of decoy messages is done by using static data. The shortcoming of this paper was that it did not provide a systematic evolution of the model.

2.4 Hybrid Models

Many researchers make use of the hybrid models in which a combination of any encryption algorithm or hashing is used with honey encryption. One such was (Moe and Win, 2018) who added salting and hashing in order to improve the security as well as time complexity of the algorithm, better than those which is used before. Different servers were used, one was the data server in which all the data like personal information is stored whereas the second server was the honey checker in which honeywords are checked with an entered password and if it matched with any sweet words, an incident is recorded and send to the admin. To improve the encryption (Burgess, n.d.) make use of the RSA algorithm in order to improve security, this method demonstrated better encryption and also provide brute force techniques
(Sahu and Ansari, 2017), combines honey encryption with Blowfish and Advanced Encryption Standard to safeguard the system against attacks by the brute force of any kind. It is made obvious that there are two processes to honey encryption, and they are all interdependent. Another approach by (Shamini et al., 2017), makes use of such a dedicated

server that the researcher used to present a false request to the user. He created an e-commerce site to illustrate the functioning whereby if the consumer repeatedly attempts to get a shipping id with the wrong password, the honey checker identifies an instance for DoS, and the identifier will be banned and banned. As a result, the intruder will be tricked into utilizing a false portal without realizing that someone is using the incorrect shopping site and that their actions are also being recorded.
The model suggested by (Sahu and Ansari, 2017) used a combination of AES encryption with honey encryption, however, Ansari additionally uses blowfish, it is evident from his findings that blowfish have better outcomes with respect to time efficiency. When HE and blowfish encryption were applied, there was a 248-millisecond difference. In comparison to AES, which required 250 milliseconds, the blowfish performed full decryption, encryption, encoding, and decoding in just 2 milliseconds.
It is demonstrated that the functioning of blowfish is improved when small, sparsely packed information is encrypted (Dibas and Sabri, 2021). Additionally, it demonstrates that when the information is sufficiently big, AES outperforms Blowfish in terms of encryption and decryption times.
The Rivest Cipher (RC5) is utilized in the approach suggested by (Vivek Raj et al., 2020). To increase the total security of the information transmission, he implemented honey encryption for key sharing. The emphasis was placed on the RC5 rounds and overall temporal model, which increases avalanche strength and security. The author of the paper came to the following conclusion: 63% avalanche with 20 rounds. To sum up the total literature, every strategy that has been used to date is a hybrid model.
Various studies utilize RC5, Blowfish, and AES with Honey encryption in the hybrid. These papers addressed a variety of issues, including typos, storage overhead, the development of honeywords, etc. To reduce overall processing time, several researchers (Vivek Raj et al., 2020) concentrated on performance. Others, (Shamini et al., 2017) where the goal wasn't primarily to save processing time, but security focused. To illustrate originality, the model in this study uses Honey encryption and the indexed method. A fresh approach is developed using a hybrid version of the Two Fish and honey encryption.

3 Research Methodology

The general finding of the existing literature serves as the foundation for the study methodology. Evaluation of the hybrid honey encryption model with two fish is the main objective. In order to construct the secured model, the procedures Plan, Design, Code, Test, and approach are employed for each stage. The workings of honey encryption with the Twofish and AES algorithms are presented in this study. A comparative study is carried out to determine which algorithm is truly effective in terms of computational time and security.

3.1 Random Honeyword Creation Technique

Honey Encryption is basically used to protect the system from any attacker trying password attacks like brute force. Honey Encryption's main objective is the creation of honeywords. Honeywords are bogus credentials and decoys that can be used to tell if a brute-force attack assault is being used. Honeywords defend against stolen credentials and give the hacker a deceptive certainty about the passwords, ultimately confusing the attacker as to which passcode is the true password.
A honeyword can be any password that is saved in a system, and honeywords can be created through honeyword creation algorithms. To determine if the supplied credentials are valid or

fake, utilize the honey checker. It maintains a list of authentic and false passwords. The honey checker can sound an alert and record information when an odd incident is discovered, as well as store it for future use (“Honey Encryption Algorithms - Security Combating Brute Force Attack | Rapid7 Blog,” 2017).

First of all, a password and a message are taken as input. After that, a random seed is selected which will map in two dictionary passwords to seeds and then seeds to message. After that the creation of honeywords takes place. Honeywords are created using mathematical manipulation and alteration of data which is also called password tweaking or tailing. These honeywords will act as bogus passwords along with actual passwords.

3.2 TwoFish Encryption

After the creation of honey words, the password is secured using two fish Encryption. The TwoFish algorithm is similar to Blowfish as it is a type of block encryption that divides a chunk of text into 128 bits.
There are various issues that have to be handled in the method. The text ought to be 128 bits long, divided into four sections, all of which should be 32-bit and use tiny convention techniques. The very first two portions will be written, and the third and final portion will be left. Bit-XOR input will be carried out using four essential components known as whitening.
The method will go through 16 iterations because two fish are utilizing the Feistel network. Other function, which uses four S-boxes, MD5 matrices, and "IPM (pseudo-Hadamard transform)," will be carried out. The stream lightening and real encrypting will be performed afterward (Jintcharadze and Iavich, 2020). Figure 3 below depicts the Twofish encryption model.

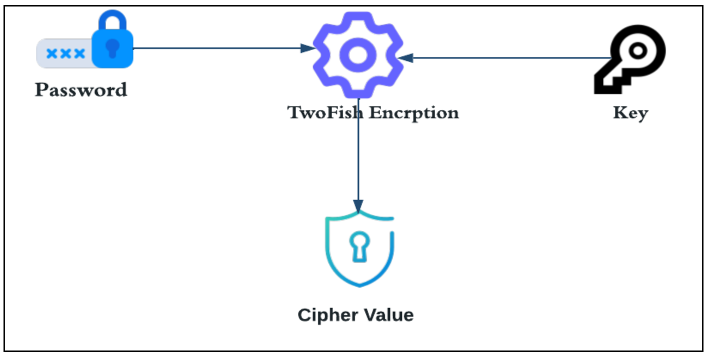


Figure 3: TwoFish encryption model

3.3 Decryption and Message Retrieval

After storing the message and encrypted password the password is decrypted first (AES or TwoFish Decryption) and then checked with the original password as well as honey words. There are three scenarios that could take place. The decryption is illustrated in figure 4 below. The message retrieval depends on the password provided by the user to extract the message. There are three possibilities that can happen:

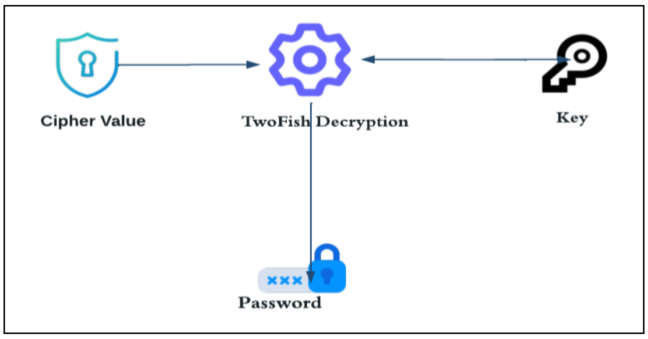


Figure 4: TwoFish decryption model

Possibility 1 - If the Password matches the original password.
A genuine message will appear if the password is accurate.
Possibility 2 - If the Password matches with any of the honey words.
If the password matched the honeyword, an alarm will be sent, and any false messages that resemble the original messages will be displayed (not the original message).
Possibility 3 - If Password incorrect (Password neither matches to honeyword or).
With the wrong password, there won't be a response (Nguyen, 2022).

4 Design Specification

The design methodology in this research aims to build a simple, implementable approach for honey word production, and it has DTE that makes it somewhat user-friendly. The suggested technique serves two purposes.
Firstly, TwoFish encryption is used to secure the password first.
Additionally, the message is protected from brute force attacks using honey encryption. Furthermore, this paper compares the use of honey encryption using AES or TwoFish.

4.1 Proposed Algorithm: Honey2Fish

1. Start
2. Take username, password, key size, and message (State of US) as input.
3. Initialize the honey Encryption.
4. Select any random seed value.
5. Map “Passwords to Seeds” And “Seeds to Messages”.
6. Create honeywords (sweet words) with manipulation of password digit tweaking and tailing techniques.
7. Initialize TwoFish Encryption. Check the block size of 16 bytes. If the block size is not multiple of 16 add padding.
8. Encrypt the password with the help of the python library of two fish.
9. Decrypt Two fish encrypted passwords and check them with the original password.
10. If the password matches the original password, go to step 11, and if the password matches with any honeywords then go to step 12. If the password does not match any of the two, then go to step 13.
11. Print the original State corresponding to the input provided.
12. Print any fake state other than the actual state and also create alert that says intruder detected.
13. Print the wrong password.

14. Stop.
Figure 5 shows the proposed hybrid model.

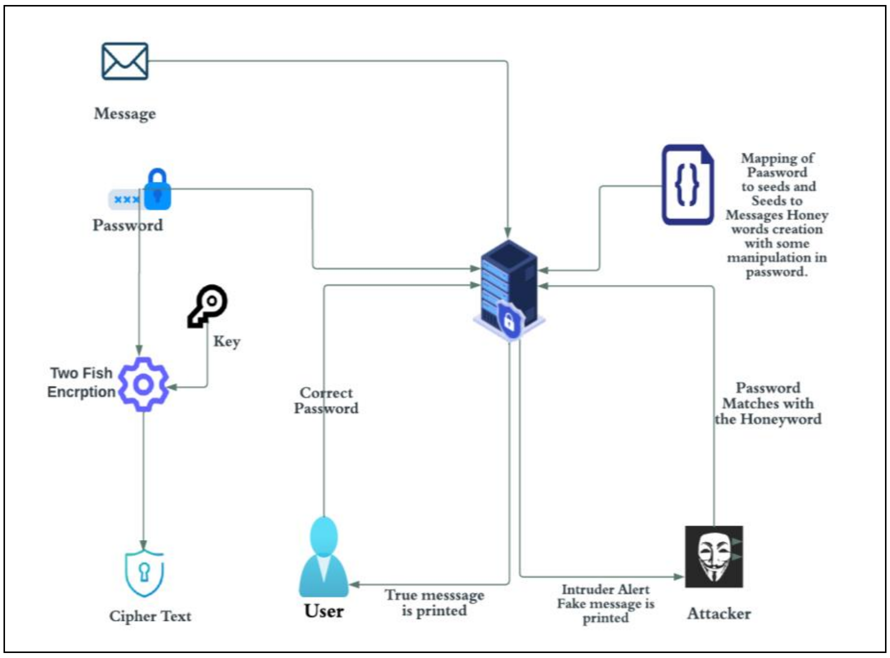

Message
Password
Key
Two Fish Encrption
Cipher Text
Correct Password
User
True messsage is printed
Intruder Alert Fake message is printed
Attacker
Password Matches with the Honeyword
Mapping of Paasword to seeds and Seeds to Messages Honey words creation with some manipulation in password.

Figure 5: Honey2Fish model

5 Implementation

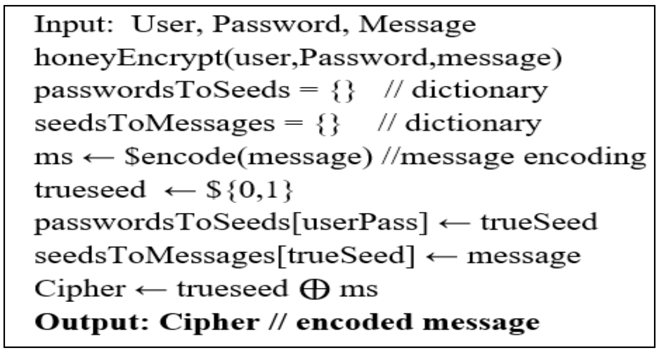
This section discussed the implementation phase. A windows operating machine having the details of the following systems:

Processor	Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz
RAM	8.00 GB
System type	64-bit operating system, x64-based processor

Python 3.0 is used as a programming language. Visual studio code is used as a code editor. The code is divided into a number of different python files. The main file calls all the other files including honey,py (Honey encryption file), twofish.py (Twofish encryption file), Aes.py (AES encryption file), etc.
Whenever an input is given, first of all, passwords along with the message are sent to the honey.py. In honey encryption, a random seed is selected, and a hardcoded dictionary is created which consists of US states (for demonstration purposes). Another dictionary that contains real passwords and honey words (manipulated passwords) is created in order to protect the real password.

Honey Encryption: **Cipher $\leftarrow$ sk $\oplus$ sm**
After creating the honeywords and mapping the password to seeds and seeds to the message, encoding takes place (Jordan, 2021).

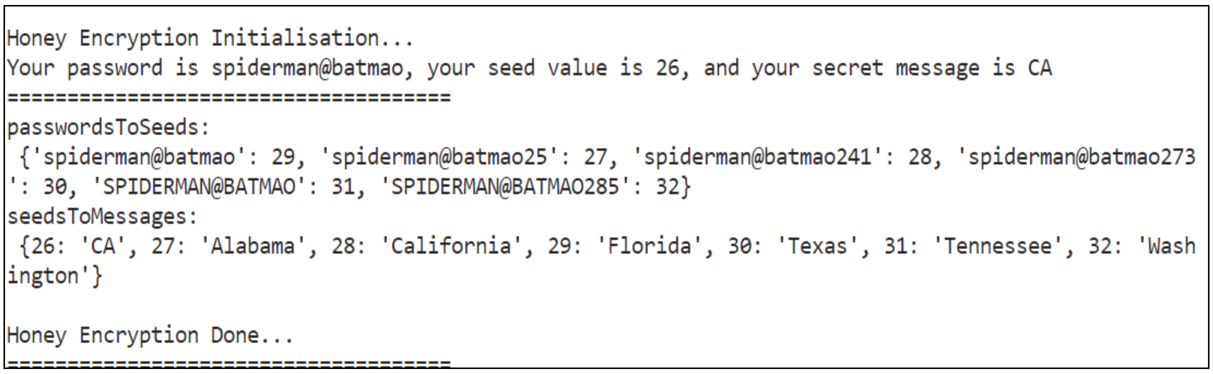
Pseudocode:



```
Input:  User, Password, Message
honeyEncrypt(user,Password,message)
passwordsToSeeds = {}   // dictionary
seedsToMessages = {}    // dictionary
ms ⟵ $encode(message) //message encoding
trueseed  ⟵ ${0,1}
passwordsToSeeds[userPass] ⟵ trueSeed
seedsToMessages[trueSeed] ⟵ message
Cipher ⟵ trueseed ⊕ ms
Output: Cipher // encoded message
```

Figure 6 : Shows PseudoCode of Honey Encryption

The output of the same is shown in Figure 7.

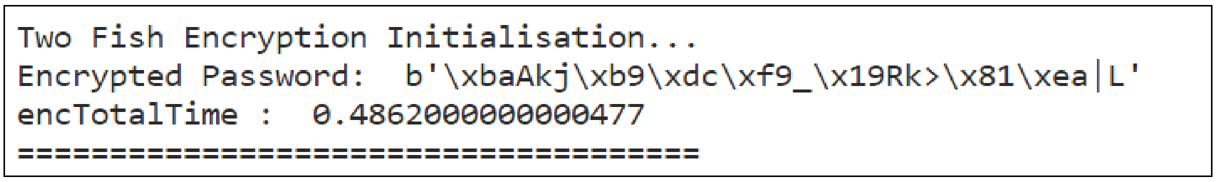


```
Honey Encryption Initialisation...
Your password is spiderman@batmao, your seed value is 26, and your secret message is CA
=====================================
passwordsToSeeds:
 {'spiderman@batmao': 29, 'spiderman@batmao25': 27, 'spiderman@batmao241': 28, 'spiderman@batmao273
': 30, 'SPIDERMAN@BATMAO': 31, 'SPIDERMAN@BATMAO285': 32}
seedsToMessages:
 {26: 'CA', 27: 'Alabama', 28: 'California', 29: 'Florida', 30: 'Texas', 31: 'Tennessee', 32: 'Wash
ington'}

Honey Encryption Done...
=====================================
```

Figure 7: Output of Honey Encryption

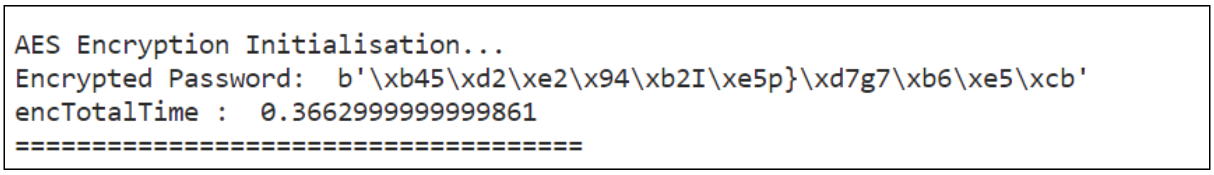
Twofish Encryption: Two arguments password, the key size is sent from the main function. Then, TwoFish () method is invoked, and passwords are set as an argument in bytes as a parameter, called cipher T. The password is then encrypted in equal-sized portions or chunks and a cipher is obtained (O, 2021).



```
Two Fish Encryption Initialisation...
Encrypted Password:  b'\xbaAkj\xb9\xdc\xf9_\x19Rk>\x81\xea|L'
encTotalTime :  0.4862000000000477
=====================================
```

Figure 8: Output of TwoFish Encryption

AES Encryption: For AES 128bit encryption an initialization vector is chosen randomly. A password and key acted as arguments to the ENC_AES method. The password is then encrypted with CBC mode.



```
AES Encryption Initialisation...
Encrypted Password:  b'\xb45\xd2\xe2\x94\xb2I\xe5p}\xd7g7\xb6\xe5\xcb'
encTotalTime :  0.3662999999999861
=====================================
```

Figure 9: Output of AES Encryption

Decryption and Message Retrieval: The message is initially decrypted from AES or Twofish, depending on the encryption used, in the decryption process. After that the honey word decryption takes place. After that, a password is provided by the user to retrieve the message.

Honey decoding: **Message $\leftarrow$ sk $\oplus$ Cipher**

The original password is verified after the password has been decrypted and the message has been decoded. Three possibilities exist: If the password is accurate, the real US state is displayed. If the password contains any honeyword, a false state of the United States is displayed; otherwise, an invalid password is displayed.

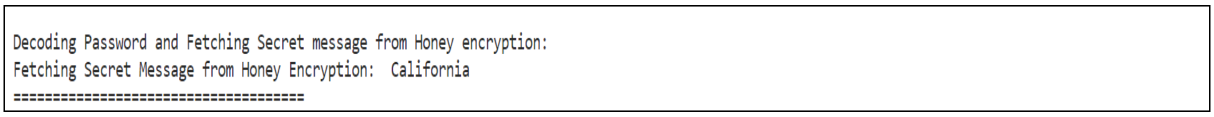
Honey Decoding Pseudo code:

```
Input: Cipher
ms ← trueseed ⊕ Cipher // ms = message encoding
message ← decode(ms)
Output: Message
```

Figure 10: Pseudo code of message decoding

Possibility 1:

The password is correct.



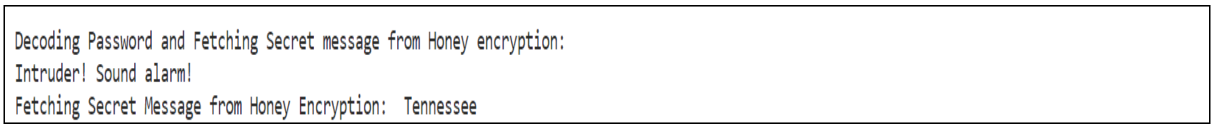
```
Decoding Password and Fetching Secret message from Honey encryption:
Fetching Secret Message from Honey Encryption:  California
===================================
```

Figure 11: Shows the correct message when the password is correct

It can be observed that the message (abbreviations of states) by the user was CA and the corresponding state California is printed.

Possibility 2:

Password matches with honeyword.



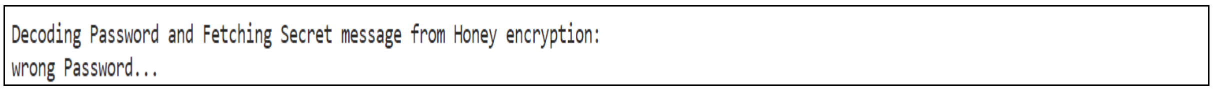
```
Decoding Password and Fetching Secret message from Honey encryption:
Intruder! Sound alarm!
Fetching Secret Message from Honey Encryption:  Tennessee
```

Figure 12: Shows the fake message when the password matches to honeyword.

When the password provided by the user matched any honeyword any fake state id is printed in this case it is Tennessee.

Possibility 3:

The password is incorrect.



```
Decoding Password and Fetching Secret message from Honey encryption:
wrong Password...
```

Figure 13: Shows the no message as the password is incorrect

When the password is incorrect only the message "wrong password…" is printed.

To summarise the overall possibilities it can be seen that if the password matches by honeyword then a fake message will be printed which will confuse the attacker about the real message.

.

6 Evaluation

This section illustrates the evaluation of the proposed model. The evaluation is done by calculating the total execution time, encryption time, and decryption time. The performance analysis is done by calculating the throughput. The security is evaluated by the avalanche effect.The two-fish algorithm is compared with the AES algorithm.

6.1 Analysis of different data sizes

The size of the password is changed and the variation in the total encryption and decryption time is calculated. The algorithm discussed is Twofish. The graph in figure 15 shows the encryption as well as decryption time. It is evident from the graph that the time increases if the size of the password is increased.

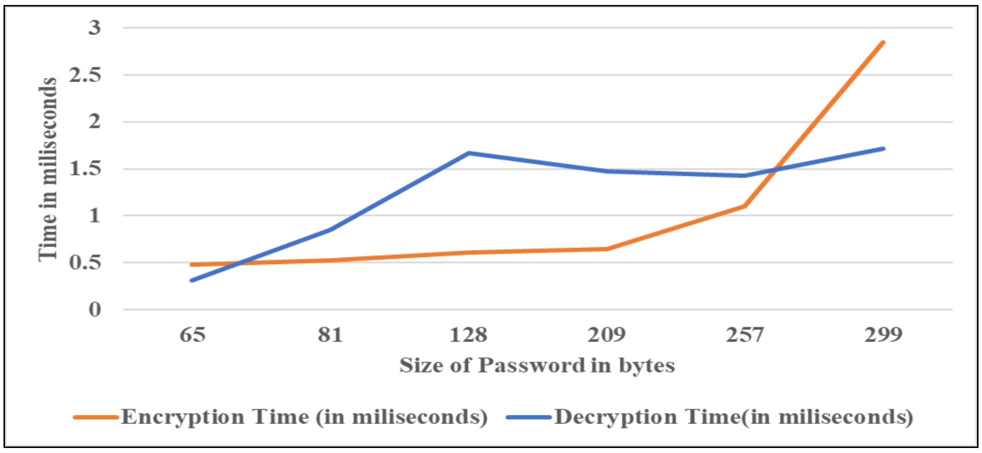


Figure 15: Graph of the encryption and decryption time when the size of the password is changed

Now the analysis of the total execution size.

Total Execution Time = Honey encryption+Twofish Encrption Time + Twofish decryption Time + Message Retrieval

The graph in figure 16 shows the total execution time with variations n the size of the password.

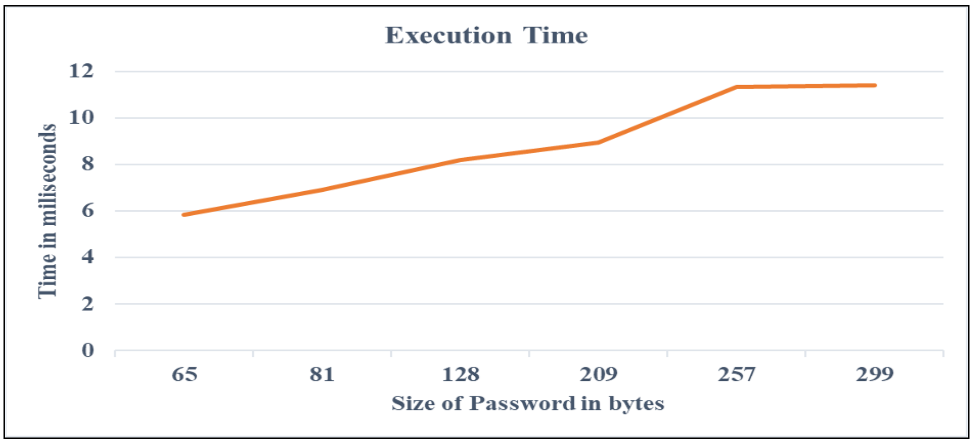


Figure 16: Graph of the total execution time when the size of the password is changed.

Now the analysis of the Through Put when the length of the password is changed.

Throughput = Size of password in MB / Total Execution Time in seconds

Size in bytes	Size in MB	Execution Time (in milliseconds)	Execution Time (in seconds)	Throughput
65	0.000065	5.8213	0.0058213	0.011165891
81	0.000081	6.8987	0.0068987	0.011741343
128	0.000128	8.17	0.00817	0.015667075
209	0.000209	8.9488	0.0089488	0.023355087
257	0.000257	11.3062	0.0113062	0.022730891
299	0.000299	11.3884	0.0113884	0.026254786

Table 2: Shows the throughput values when the length of the password is changed.

6.2 Analysis of AES and TwoFish

The comparison of AES and Twofish encryption is shown in the graph represented in figure 17. The Variation in execution time as the size of the password varied.

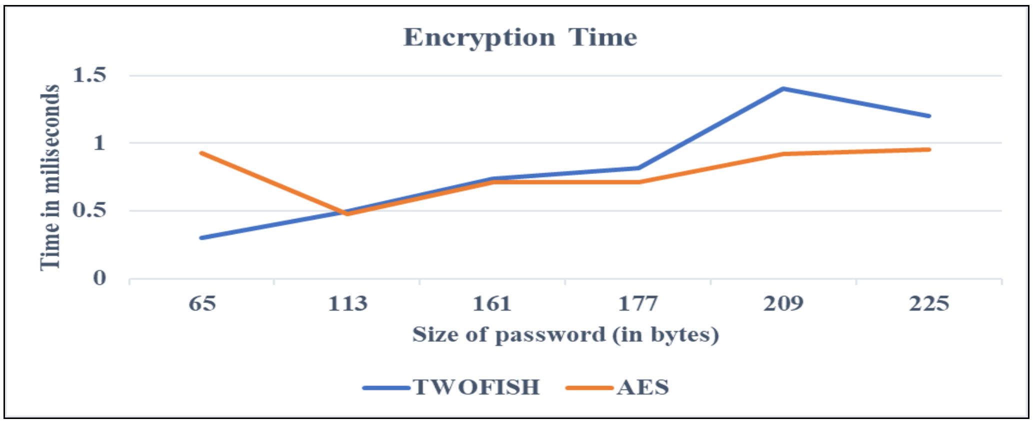


Figure 17: Graph of the encryption time of AES and Twofish

The comparison of AES and Twofish Decryption is shown in the graph in figure 18 below. The Variation in execution time as the size of the password varied is shown.

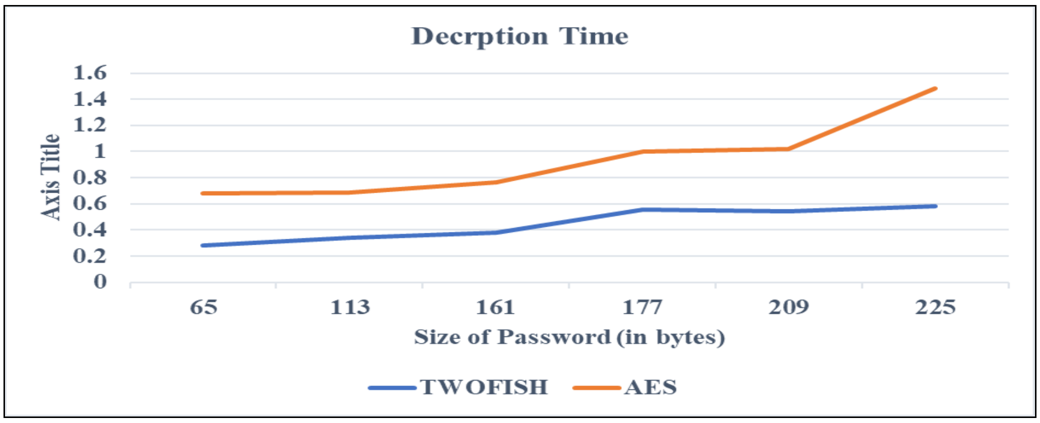
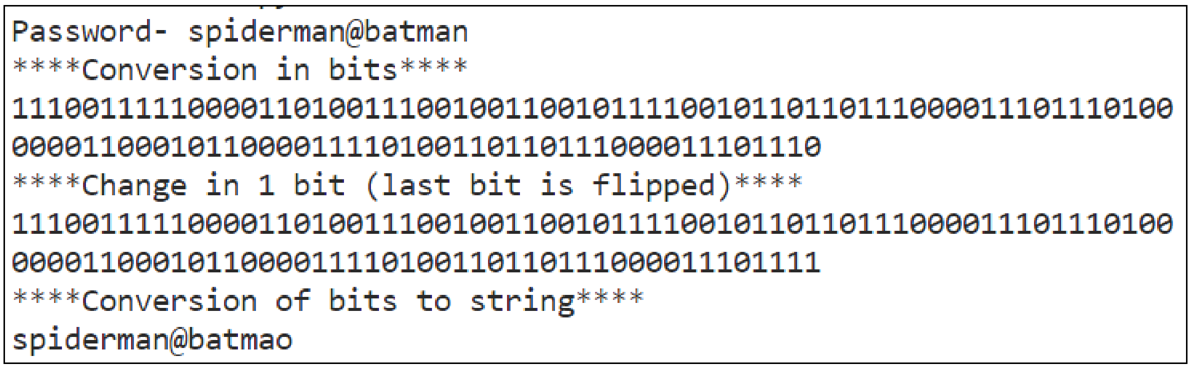


Figure 18: Graph of the decryption time of AES and Twofish

6.3 Analysis of avalanche effect

The phrase "avalanche effect" in crypto is used to evaluate the security of the model. Any cryptosystem would benefit from having the avalanche effect as one of its desirable properties. It is defined as a small modification in input (preferably 1 bit) that shows how much effect the output.
The formula of the Avalanche Effect
Avalanche effect = (Total number of bits flipped) / (Total number of bits)

For the calculation of the avalanche effect, the password changes into bits, and then one bit is changed to get the respective encrypted password. After getting these encrypted passwords are converted into bits and then the number of changed bits is calculated. The figure below shows the bit conversion and bit flipping.



```
Password- spiderman@batman
****Conversion in bits****
1110011111000011010011100100110010111100101101101110000111011110100
0000110001011000011110100110110111000011101110
****Change in 1 bit (last bit is flipped)****
1110011111000011010011100100110010111100101101101110000111011110100
0000110001011000011110100110110111000011101111
****Conversion of bits to string****
spiderman@batmao
```

Figure 19: Shows the bit conversion.

The below table shows the avalanche effect with respect to different plaintext(password):

Plain Text (Password)	Size in bytes	Changed Plaintext	Cipher Text (Before Change)	Cipher Text (After Change)	Avalanche Effect
Tiger	54	Tige**s**	b'\x0eRM\xc9\xcbK\xfb\xd1\xf0X\x00\xd5\xbd\xe0>\x88'	b':d$\x01\xb5\xde\xccN+\xd0_\xd2Wl\x93\xeb'	48.54%
Mangoapple	59	Mangoappl**d**	b'Mf\x8d\xb91\xce\xa1\x9d\x95L\xef\xb8\xc2@QE'	b'k\x11\xc6\x9b\xe0\x8c\x04J\xb5\xcf\xca}?\x8e-\xa9'	50.58
spiderman@batman	65	spiderman@batma**o**	b'\x10\xe7\xb8\xbb\xb9j\xbd\xf1\xad\xcc\x87T\x88U\t\x06'	b'\xbaAkj\xb9\xdc\xf9_\x19Rk>\x81\xea|L'	53.09%
jdhfbodvbnkdjvbdjkwd	69	jdhfbodvbnkdjvbdjkw**e**	b"\xd7lu\x91\xd8\xb4+\xe7W$\xb3\xf1M}P\x8c\x8d\x92\xee\xf8.\xbe\xb7/\x88\x98\x0e'0\xcbg\x02"	b'\xd7lu\x91\xd8\xb4+\xe7W$\xb3\xf1M}P\x8c\x86K\x1bo\xee\x9e\x96\x99\xf1y\xe2$\x0f\xb9lb'	53.61%

Table 3: Shows the avalanche effect.

The graph shown in figure 20 below depicts the variation in the avalanche effect when the size of the password is changed.

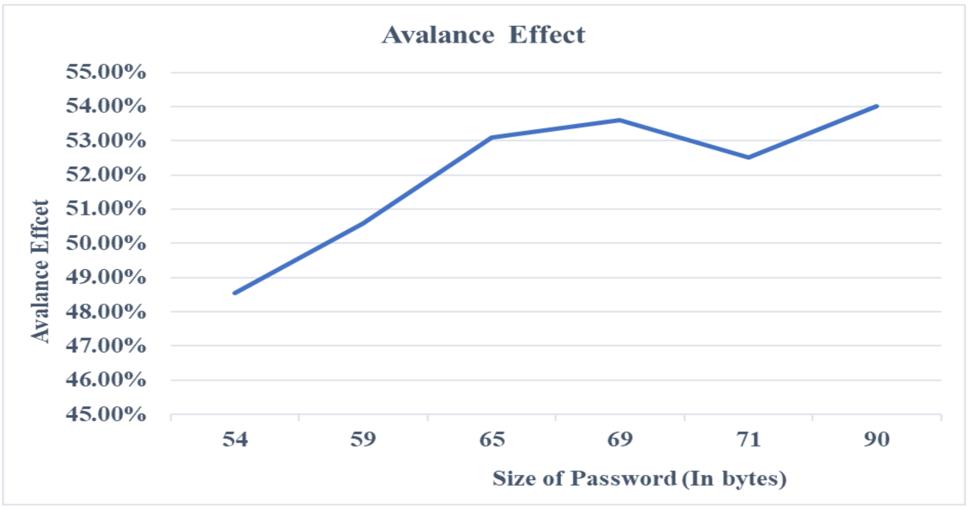


Figure 20: Graph of the avalanche effect when the size of the password is changed.

6.4 Discussion

From the proposed model using TwoFish with Honey Encryption and already implemented model RC2 with Honey Encryption by (Vivek Raj et al., 2020), it is evident that the encryption time taken by RC2 is much higher than that of two fish. The method for generating the honeyword in this study is relatively easy to implement used by (Sahu and Ansari, 2017). Twofish also results in better overall execution time than used with RC2. The execution time is also calculated when the size of the password is changed. It can be seen in the evaluation section that the encryption time and decryption time depend upon the size of the password. If the password size is increased, then the time taken also increases. The security is evaluated by calculating the avalanche effect. The password-based encryption (PBE) techniques now used to safeguard sensitive information are weak against brute-force attacks because the intruder may tell if the secret, they guess is accurate or not simply by examining the outcomes of the ciphertext. The honey encryption method is a defense against such a weakness. Honey encryption is a technique used to protect the system from an attacker by providing the attacker a valid looking information that is incorrect.
The proposed method has Avalanche Effect slightly better compared to existing models. As the avalanche effect is better, it can be said that the security of the proposed hybrid model is better. The throughput is also calculated which is also compared with different password lengths. The problem of complicated DTE is addressed in this proposed model. It is evident from the proposed model that performance depends upon the size of the password. The security is checked using the avalanche effect. A comparison of Twofish and AES is also carried out in which we can conclude that AES required more decryption time than Twofish

7 Conclusion and Future Work

Private information needs to be adequately safeguarded to prevent loss from leaks and abuse. The Honey encryption is a combination of DTE and honeyword creation. In the proposed model a simple honeyword creation technique is used which is easy to implement. To provide an extra layer of security twofish encryption is being implemented. The message is secured using honey encryption and the password is secured by Twofish encryption. To conclude it is evident from the above evaluation that twofish is better than RC2 when it comes to performance. The security is also better than RC2 as we have compared the avalanche effect. When compared with the AES, the proposed model with Twofish shows slightly higher encryption time as it also depends on the configuration of the system.

The presented model's potential future applications include combining several algorithms like 3DES, RSA, etc. To increase the system's overall security, hashing can also be used. The proposed honey encryption can be used in real-life scenarios, such as credit card OTP security and fraud detection websites.

8 References

A Novel Hybrid Encryption Method Based on Honey Encryption and Advanced DNA Encoding Scheme in Key Generation [WWW Document], n.d. URL https://www.scirp.org/journal/paperinformation.aspx?paperid=119724 (accessed 13/12/2022).

AlMuhanna, A., AlFaadhel, A., Ara, A., 2022. Enhanced System for Securing Password Manager Using Honey Encryption, in 2022 Fifth International Conference of Women in Data Science at Prince Sultan University (WiDS PSU). Presented at the 2022 Fifth International Conference of Women in Data Science at Prince Sultan University (WiDS PSU), pp. 150–154. https://doi.org/10.1109/WiDS-PSU54548.2022.00042

Bangera, S., Billava, P., Naik, S., 2020. A Hybrid Encryption Approach for Secured Authentication and Enhancement in Confidentiality of Data, in 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC). Presented at the 2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC), pp. 781–784. https://doi.org/10.1109/ICCMC48092.2020.ICCMC-000145

Burgess, J., n.d. Honey Encryption.

Chakraborty, N., Mondal, S., 2017. On designing a modified-UI based honeyword generation approach for overcoming the existing limitations. Computers & Security 66, 155–168. https://doi.org/10.1016/j.cose.2017.01.011

Chen, H.-C., Wijayanto, H., Chang, C.-H., Leu, F.-Y., Yim, K., 2016. Secure Mobile Instant Messaging key exchanging protocol with One-Time-Pad substitution transposition cryptosystem, in: 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). Presented at the 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 980–984. https://doi.org/10.1109/INFCOMW.2016.7562224

Dibas, H., Sabri, K.E., 2021. A comprehensive performance empirical study of the symmetric algorithms:AES, 3DES, Blowfish and Twofish, in: 2021 International Conference on Information Technology (ICIT). Presented at the 2021 International Conference on Information Technology (ICIT), pp. 344–349. https://doi.org/10.1109/ICIT52682.2021.9491644

Erguler, I., 2016. Achieving Flatness: Selecting the Honeywords from Existing User Passwords. IEEE Transactions on Dependable and Secure Computing 13, 284–295. https://doi.org/10.1109/TDSC.2015.2406707

Honey Encryption Algorithms - Security Combating Brute Force Attack | Rapid7 Blog [WWW Document], 2017. . Rapid7. URL https://www.rapid7.com/blog/post/2017/05/03/honey-encryption-algorithms-security-combating-brute-force-attack/ (accessed 13/12/2022).

Jaeger, J., Ristenpart, T., Tang, Q., 2016. Honey Encryption Beyond Message Recovery Security, in: Fischlin, M., Coron, J.-S. (Eds.), Advances in Cryptology – EUROCRYPT 2016, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp. 758–788. https://doi.org/10.1007/978-3-662-49890-3_29

Jintcharadze, E., Iavich, M., 2020. Hybrid Implementation of Twofish, AES, ElGamal and RSA Cryptosystems, in: 2020 IEEE East-West Design & Test Symposium (EWDTS).

Presented at the 2020 IEEE East-West Design & Test Symposium (EWDTS), pp. 1–5. https://doi.org/10.1109/EWDTS50664.2020.9224901

Jordan, K., 2021. Honey Encryption. smucs. URL https://medium.com/smucs/honey-encryption-e56737af081c (accessed 13/12/2022).

Juels, A., Rivest, R.L., 2013. Honeywords: making password-cracking detectable, in: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, CCS '13. Association for Computing Machinery, New York, NY, USA, pp. 145–160. https://doi.org/10.1145/2508859.2516671

Moe, K.S.M., Win, T., 2018. Enhanced Honey Encryption Algorithm for Increasing Message Space against Brute Force Attack, in: 2018 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). Presented at the 2018 15th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), pp. 86–89. https://doi.org/10.1109/ECTICon.2018.8620050

mohammed, S., KURNAZ, S., Mohammed, A.H., 2020. Secure Pin Authentication in Java Smart Card Using Honey Encryption, in: 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). Presented at the 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), pp. 1–4. https://doi.org/10.1109/HORA49412.2020.9152936

Nguyen, V., 2022. Honey Encryption.

Nirmalraj, T., Jebathangam, J., 2022. A Password Secure Mechanism using Reformation-based Honey Encryption and Decryption, in: 2022 International Conference on Inventive Computation Technologies (ICICT). Presented at the 2022 International Conference on Inventive Computation Technologies (ICICT), pp. 214–220. https://doi.org/10.1109/ICICT54344.2022.9850868

O, K., 2021. Python TwoFish Encryption. DevRescue. URL https://devrescue.com/python-twofish-encryption/ (accessed 13/12/2022).

Omolara, A.E., Jantan, A., Abiodun, O.I., Poston, H.E., 2018. A Novel Approach for the Adaptation of Honey Encryption to Support Natural Language Message. Hong Kong.

Pagar, V.R., Pise, R.G., 2017. Strengthening password security through honeyword and Honeyencryption technique, in: 2017 International Conference on Trends in Electronics and Informatics (ICEI). Presented at the 2017 International Conference on Trends in Electronics and Informatics (ICEI), pp. 827–831. https://doi.org/10.1109/ICOEI.2017.8300819

Sahu, R., Ansari, M.S., 2017. Securing Messages from Brute Force Attack by Combined Approach of Honey Encryption and Blowfish.

Sahu, S., 2020. PROVIDING INFORMATION SECURITY USING HONEY ENCRYPTION. Adv. Math., Sci. J. 9, 8249–8258. https://doi.org/10.37418/amsj.9.10.54

Shamini, P.B., Dhivya, E., Jayasree, S., Lakshmi, M.P., 2017. Detection and avoidance of attacker using honey words in purchase portal, in: 2017 Third International Conference on Science Technology Engineering & Management (ICONSTEM). Presented at the 2017 Third International Conference on Science Technology Engineering & Management (ICONSTEM), pp. 260–263. https://doi.org/10.1109/ICONSTEM.2017.8261290

Shen, C., Yu, T., Xu, H., Yang, G., Guan, X., 2016. User practice in password security: An empirical study of real-life passwords in the wild. Computers & Security 61, 130–141. https://doi.org/10.1016/j.cose.2016.05.007

Tan, S.-F., Lo, K.-M.C., Leau, Y.-B., Chung, G.-C., Ahmedy, F., 2021. Securing mHealth Applications with Grid-Based Honey Encryption, in: 2021 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET). Presented at the 2021 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET), pp. 1–5. https://doi.org/10.1109/IICAIET51634.2021.9573645

Vivek Raj, K., Ankitha, H., Ankitha, N.G., Kanthi Hegde, L.S., 2020. Honey Encryption based Hybrid Cryptographic Algorithm:A Fusion Ensuring Enhanced Security, in: 2020 5th International Conference on Communication and Electronics Systems (ICCES). Presented at the 2020 5th International Conference on Communication and Electronics Systems (ICCES), pp. 490–494. https://doi.org/10.1109/ICCES48766.2020.9137849