

# User Authentication & Protection Using Decoy Technology & Honeyword

Jigar Bipin Chheda<sup>1</sup>, Sujay Shankar Pol<sup>2</sup>, Vaishnavi Shivji Joshi<sup>3</sup>

<sup>1-3</sup>Students, Dept. of Computer Engineering, Pillai Hoc College of Engineering & Technology, Rasayani, Maharashtra, India

\*\*\*

**Abstract** - We recommend a generally exact and best technique for improving the security of hashed passwords by giving of extra "honeywords" (bogus passwords) related with every client's record. An intruder who needs to takes a document of hashed passwords and modifies the hash work can't tell on the off chance that he has discovered the secret phrase or a honeyword. the "Honeychecker" can recognize the client secret phrase from honeywords for the login schedule, and will set off an alert if a honeyword is use. we propose an elective methodology that select the honeywords (fake pass) from existing client secret phrase in the framework to give practical honeywords, a completely level honeyword age strategy and furthermore to decrease the capacity cost.

**Key Words:** Honeywords, Intruder, Honeychecker, Bogus, Password, Hash

## 1.INTRODUCTION

Revelation of secret password might be a serious security issue that has influenced various clients and organizations like Yahoo, LinkedIn and Adobe, since spilled passwords make the clients focus of the numerous conceivable digital assaults. These new occasions have exhibited that the feeble secret word stockpiling strategies are right now in situ on numerous sites. for example, the LinkedIn passwords were utilizing the SHA-1 calculation without a salt and comparably the passwords inside the eHarmony framework were likewise put away utilizing unsalted MD5 hashes. In fact, when a secret word record is taken, by utilizing the secret phrase breaking strategies very much like the calculation of Weir et al. it's not difficult to catch a large portion of the plaintext passwords. In this regard, there are two issues that should be considered to beat these security issues: First, passwords should be ensured by avoiding potential risk and putting away with their hash esteems registered through salting or another perplexing components. Hence, for a foe it should be difficult to upset hashes to gather plaintext passwords. The subsequent point is that a protected framework ought to identify whether a secret word exposure episode occurred or to not make proper moves. during this investigation, we have practical experience in the last issue and focus on counterfeit passwords or records as a direct and cost compelling answer for distinguish bargain of passwords. Honeypot is one among the techniques to spot event of a secret phrase data set break. during this methodology, the manager intentionally makes misleading passwords to bait foes and identifies a secret word

divulgence, in the event that anybody of the honeypot passwords gets utilized. during this examination, we break down the honeyword approach and gives a few comments about the wellbeing of the framework. Besides, we infer that the critical thing for this strategy is that the age calculation of the honeywords indicated they will be undefined from the legitimate passwords. In this manner, we propose a substitution approach that utilizes passwords of different clients inside the framework for honeyword sets, for example practical honeywords are given. Besides, this framework likewise lessens the capacity cost. Fundamentally, a direct however smart thought behind the investigation is that the inclusion of bogus passwords – called as honeywords – related with every client's record. At the point when an enemy gets the password rundown, she recuperates numerous secret phrase possibility for each record and the person can't make certain about which word is veritable. Subsequently, the broke password documents are frequently recognized by the manager if a login endeavor is done with a honeyword by the foe.

### 1.1 Honeywords

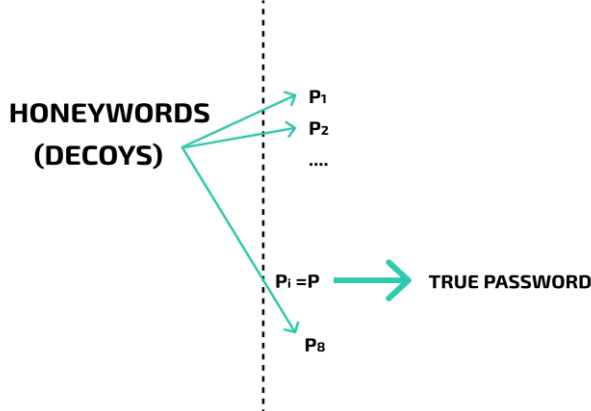
At the point when a foe gets a password document he/she may get confounded between the secret word, which one is a right. It gives Basically, it's the addition of bogus passwords identified with every client's record. At the point when a foe gets the password run down she recuperates numerous password possibility for each record and she or he can't make certain about which word is authentic. Consequently, the broke secret phrase documents is identified by the PC client if a login endeavor is done with a honeyword by the foe.

Honeyword Generator calculation Gen () - The creators present a definition in light of the fact that the levelness of Gen () indicated it gauges the possibility of an enemy in picking the appropriate password from the sweet words.

Primary wording behind this framework is, in information base we store the genuine password and the honeywords in a similar record.

In below fig.1.1 we can see that P is the genuine secret word of the client where, as p1,p2,... pn are the honeywords of that client.

## ■ TERMINOLOGY



These passwords are not produced haphazardly as in the current framework. In our framework honeywords are the passwords of the other genuine clients which are doled out arbitrarily at the hour of enlistment of deception of the genuine passwords to programmer so that login to that record gets hard for the programmer.

It is not difficult to allocate an arbitrary clients passwords to specific client yet it is hard for the programmer to get the genuine password to get the passage into the client's record. After a 3 login endeavor programmer gets passage into the phony record which is made by the administrator. Administrator can transfer the phony documents to the phony record to give the figment of the genuine client account. Administrator will be advise through mail and message framework with the username who are attempting to hack that client's record. Administrator will apply the security designs further to forestall the hacking or to hinder that programmer. This exercises are dealt with by the administrator, client is uninformed of this exercises.

## 2. LITERATURE REVIEW

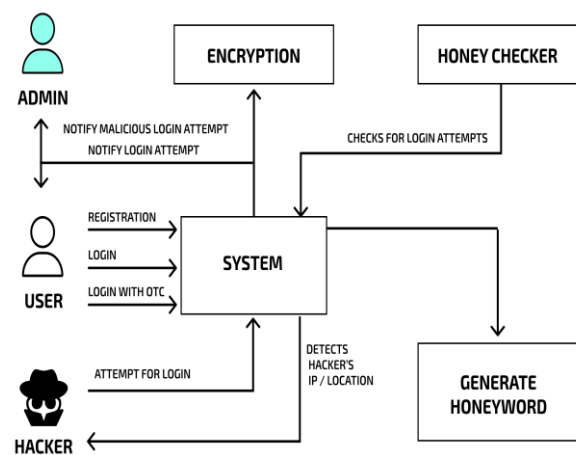
This thought has been adjusted by Herley and Florencio to shield internet banking accounts from password savage power assaults. As per the examination, for every client mistaken login endeavors for certain passwords lead to honeypot accounts, for example malevolent conduct is perceived. For example, there are 108 opportunities for a 8-digit password and let framework joins 10000 wrong password to honeypot accounts, so the enemy playing out the savage power assault multiple times bound to hit a honeypot account than the certified record. Utilization of distractions for building burglary safe was presented by Bojinov et al. in called as kamouflage. In this model, the phony secret word sets are put away with the genuine client secret phrase set to hide the genuine passwords, along these lines constraining an enemy to do a lot of online work prior to getting the right data.

Recently, Juels and Rivest have introduced the honeyword instrument to distinguish an enemy who endeavors to login with broke passwords. Fundamentally, for each username a bunch of sweet words is developed to such an extent that just a single component is the right secret phrase and the others are honeywords (imitation passwords). Subsequently, when a foe attempts to go into the framework with a honeyword, a caution is set off to tell the chairman about a secret phrase spillage.

## 3. EXISTING SYSTEM

As of late, Juels and Rivest proposed honeywords (fake passwords) to identify assaults against hashed secret word data sets. for each client account, the genuine secret phrase is put away with a few honeywords to detect pantomime. On the off chance that honeywords are chosen appropriately, a digital aggressor who takes a document of hashed passwords can't be certain if it's the first secret word or a honeywords for any record. In addition, entering with a honeyword to login will trigger a caution informing the manager a couple of secret word document penetrate. To the detriment of quickening the capacity necessity by multiple times, the creators present a clear and compelling answer for the identification of secret phrase record exposure occasions. during this examination, we investigate the honey word framework and present a few comments to highlight conceivable flimsy parts. Likewise, we prompt another methodology that chooses the honey words from existing client passwords inside the framework in order to supply reasonable honey words-a brilliantly level honeyword age strategy and furthermore proportional back stockpiling cost of the honey word conspire.

## 4. PROPOSED SYSTEM



Proposed model is as yet dependent on utilization of honeywords to distinguish password cracking. However, instead of existing user's password we are generating the

honey words and storing them in the password files for more realistic.

Every Registered user would have multiple fake entries in the database at the time of registration, which would create confusion in attackers mind as he/she would be seeing multiple realistic user entries or duplicate user entries and it would be difficult to identify the original user or the actual user.

Admin or user authentication via email that whether the user or admin has currently login to the system or not and if not logged in he/she can directly trigger the logout button and session would be logged out.

In case if some users has any memory loss problem or finding difficulties in remembering the password then he/she can login to the system using OTC (One time code) received on registered email only.

#### 4.1 Honeyword Checker Algorithm

Algorithm: Simple Algorithm

```

1.Procedure SimpleModel
2.P←real password
3.W←array(L)           L array contains n numbers
4.d←length(w)         d returns random length of honeywords
5.honeyword(j)←p.w[i]
6.   for i←0 to d(random length) of honeywords
7.       r_w(random words)←w(i) select ith number of an array
8.       i_p(insert_pass)←strlen(p) — add into last position
9.       honeyword(j)←substr_replace(p, r_w, i_p);
10.   end for
11.end procedure
    
```

#### Inputs:

1. T fake user accounts (honey pots)
2. Index value between [1;N],  
Index list, which is not previously assign to user

#### Procedure:

Understanding Password Database Compromises

#### Step 1: Honey pots creation: fake user account

a. For each record honey list set is made like  
 $X_i = (x_{i,1}; x_{i,2}; \dots; x_{i,k})$ ; one of the elements in  $X_i$  is the correct index (sugar index) as  $c_i$

b. create two password file file F1 and file F2  
 F1 Store username and honyindex set  $\langle hui, x_i \rangle$  Where hui is honey pot account

F2 keeps the index number and the corresponding hash of the password (create the hash of the password),  $\langle c_i; H(p_i) \rangle$

#### Step 2: Generation of Honeyindex set

In Step 1 we insert honey index set in file F1 but don't know how to create that We use honey index generator algorithm  
 $Gen(k; SI) \rightarrow c_i; X_i$  Generate  $X_i$

- a. Select  $x_i$  randomly choosing  $k-1$  numbers from  $SI$  and furthermore arbitrarily picking a number  $c_i$   $SI$ .
- b.  $u_i; c_i$  pair is conveyed to the honeychecker and F1, F2 files are refreshed.

#### Step 3: Honey checker

Set:  $c_i, u_i$  Sets right password record  $c_i$  for the client  $u_i$   
 Check:  $u_i, j$  Checks whether  $c_i$  for  $u_i$  is equivalent to given  $j$ .  
 Returns the outcome and if fairness doesn't hold, informs framework a honeyword circumstance.

### 5. WORKING

#### 5.1 Initialization

Firstly, T fake user accounts are created with their passwords. Also an index value between [1, N], but not used before it is assigned to every honeypot randomly. Then k 1 numbers are randomly selected from the index list and for every account a honeyindex set is constructed like  $X_i = (x_{i,1}; x_{i,2}, \dots, x_{i,k})$ ; one in every of the weather in  $X_i$  is that the correct index (sugar index) as  $c_i$ . Presently, we utilize two secret word documents as F1 and F2 inside the fundamental worker: F1 stores username and honeyindex set,  $\langle hui, X_i \rangle$  sets where hui indicates a honeypot account. Note that every entry has two elements. the primary one is that the username of the account and also the second element is honeyindex set for the respective account. Likewise, the table is arranged one after another in order by the username field. On the opposite hand, F2 keeps the fact and also the corresponding hash of the password.

In this case, each entry within the table has two elements. The primary element is that the sugar index of the account and also the second is that the hash of the corresponding password. Notice that the table is sorted in line with the index values. Allow  $SI$  to mean the list segment and  $SH$  address the relating secret phrase hash section of F2.

**Table -1:** Example Password File F1 for the Proposed Model

Username	Honeyindex Set
SujayPol	(93, 16626, ..., 94931)
VaibhavSali	(15476, 51443, ..., 88429)
RohanP	(3, 62107, ..., 91233)
SiddT	(89, 79869, ..., 59897)
AdityaM	(1009, 23471, ..., 47623)
JigarC	(63, 51234, ..., 72382)

**Table -2:** Example Password File F2 for the Proposed Model

SI	SH
3	$H(p3)$
7	$H(p7)$
85	$H(p85)$
⋮	⋮
100000	$H(p100000)$
100004	$H(p100004)$

### 5.2 Registration

After the initialization process, system is prepared for user registration. During this phase, a legacy-UI is preferred, i.e. username and password are required from the user as ui; pi to register the system. We use the honeyindex generator algorithm. Last, periodically honey indexes of every account should be regenerated. Because the number of users within the system increases to produce uniform distribution of honey indexes across SI, fresh honeyindex set must involve numbers from this new larger list. Otherwise, passwords of newly created accounts wouldn't be used as honeywords within the system and it should provides a clue to the adversary to in guessing the right password of those new accounts. Note that inside a predictable dispersion every secret phrase is allotted as a honeyword about k occasions, on the grounds that there are N passwords however Nk honeywords are required.

### 5.3 Login Process

Our System first checks whether entered password, g, is right for the corresponding username ui. To create this, initially the Xi of the comparing ui is accomplished from the F1 file. Then, the hash values stored in F2 file for the respective indices in Xi are compared with H (g) to search out a match. If a match isn't found, then it means g is neither the proper password, nor one among the honeywords, i.e. login fails. On the opposite hand, if H (g) is found within the list, then the most server checks whether the account may be a

honeypot. If it's a honeypot, then it follows a predefined security policy against the password disclosure scenario. Notice that for a honeypot account there's no importance of the entered password is genuine or a honeyword, so it directly manages the event without communicating with the honey checker. Honey checker controls whether  $j = ci$  and returns the result to the most server. At the identical time, if it's not equal, then it assured that the proffered password may be a honeyword and adequate actions should be taken betting on the policy.

### 5.4 Purpose and Scope

- Honeywords are used in validation framework
- The primary point of undertaking is to approving if information access is allowed when strange data access is recognized.
- Mistaking the aggressor for counterfeit data.
- This secures against the abuse of the client's genuine information.
- We propose an altogether different way to deal with getting the cloud utilizing fake data innovation, that we've come to call haze figuring.
- We utilize this innovation to dispatch disinformation assaults against malevolent insiders, keeping them from recognizing the first touchy client information from counterfeit useless information.

### 5.5 Result

#### User Side:

1. Login



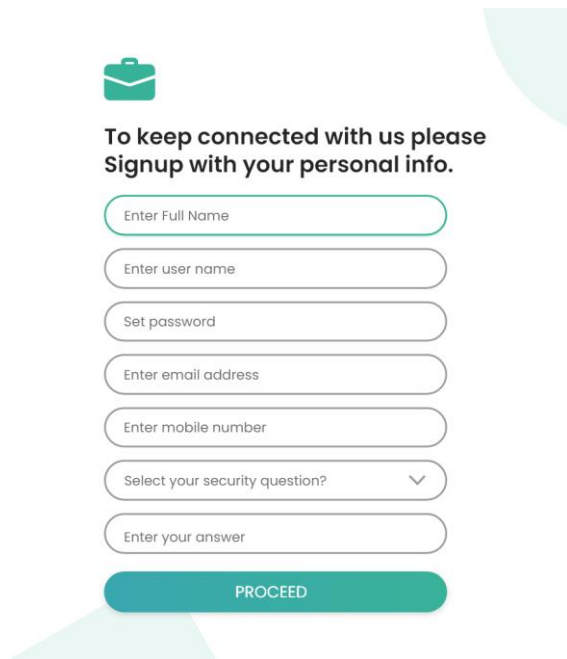
To keep connected with us please login with your personal info.



Login with OTC

LOGIN

## 2. Register



**To keep connected with us please Signup with your personal info.**

Enter Full Name

Enter user name

Set password

Enter email address

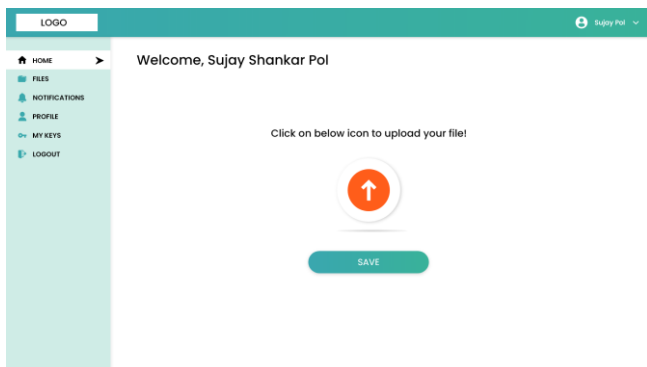
Enter mobile number

Select your security question?

Enter your answer

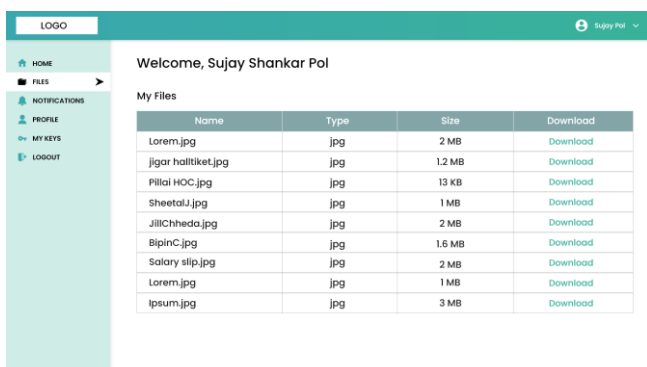
**PROCEED**

## 3. Home Page



User once logged in at his home page can upload any file that he has been working on and can be visible only to him.

## 4. Files

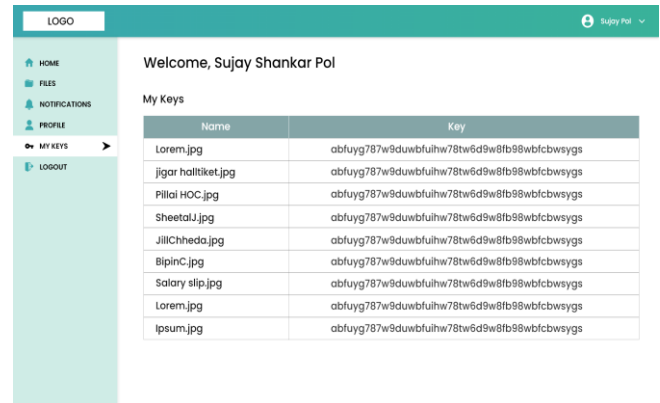


Name	Type	Size	Download
Lorem.jpg	jpg	2 MB	Download
jigar haltiket.jpg	jpg	1.2 MB	Download
Pillai HOC.jpg	jpg	13 KB	Download
Sheetal.jpg	jpg	1 MB	Download
JilChheda.jpg	jpg	2 MB	Download
BipinC.jpg	jpg	1.6 MB	Download
Salary slip.jpg	jpg	2 MB	Download
Lorem.jpg	jpg	1 MB	Download
Ipsum.jpg	jpg	3 MB	Download

For security reasons once the file is prepared the user should upload that file in database so that when the file is

required he can download it from this page using his files key which differs from file to file.

## 5. Keys

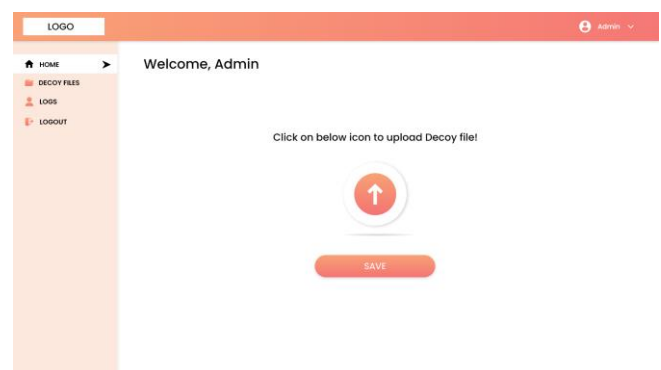


Name	Key
Lorem.jpg	abfuyg787w9duwbfuihw78tw6d9w8fb98wbfcbwsygs
jigar haltiket.jpg	abfuyg787w9duwbfuihw78tw6d9w8fb98wbfcbwsygs
Pillai HOC.jpg	abfuyg787w9duwbfuihw78tw6d9w8fb98wbfcbwsygs
Sheetal.jpg	abfuyg787w9duwbfuihw78tw6d9w8fb98wbfcbwsygs
JilChheda.jpg	abfuyg787w9duwbfuihw78tw6d9w8fb98wbfcbwsygs
BipinC.jpg	abfuyg787w9duwbfuihw78tw6d9w8fb98wbfcbwsygs
Salary slip.jpg	abfuyg787w9duwbfuihw78tw6d9w8fb98wbfcbwsygs
Lorem.jpg	abfuyg787w9duwbfuihw78tw6d9w8fb98wbfcbwsygs
Ipsum.jpg	abfuyg787w9duwbfuihw78tw6d9w8fb98wbfcbwsygs

To download the file first the user has to use his actual username and password so that he can get the key for every single file that has been uploaded by him.

## Admin Side:

1. Admin can login to this page to view the user directories and logs. The main role of admin is to upload fake files in case the hacker tries to use the user name to collect the files. Admin can only get and check the activities about user file uploads, downloads, viewed and size of the file.



## 2. Logs

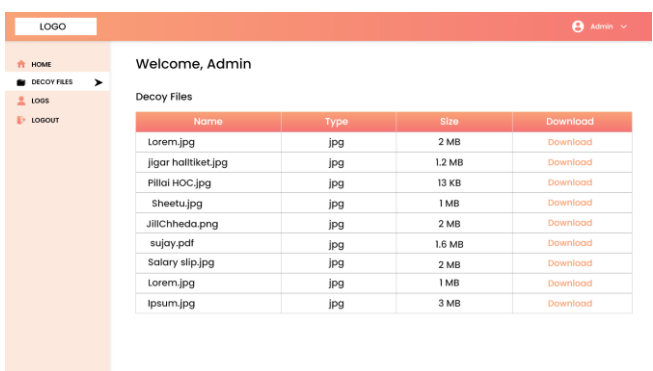
The file uploaded by admin can be accessed only by hacker, real user cannot even view those files. Admin can upload any type of fake file.



## 6. CONCLUSION

Our framework serves to administrator. Client gets moment ready when some programmer attempted to get to his record. Likewise programmer will see the rundown of fakes documents inside the framework. So he feels that he got the entrance the record. we've introduced a new way to deal with make the age calculation as close on quality by making honeywords with haphazardly picking passwords that have a place with different clients inside the framework. we've contrasted the proposed model and different techniques with significance DoS opposition, evenness, and capacity cost and convenience properties. this strategy Confuse to the aggressor with counterfeit data. This secures against the abuse of the client's unique information. We propose an outrageously extraordinary way to deal with getting the cloud utilizing bait data innovation, that we've come to call haze figuring. We utilize this innovation to dispatch disinformation assaults against malevolent insiders, keeping them from recognizing the significant delicate client information from counterfeit useless information.

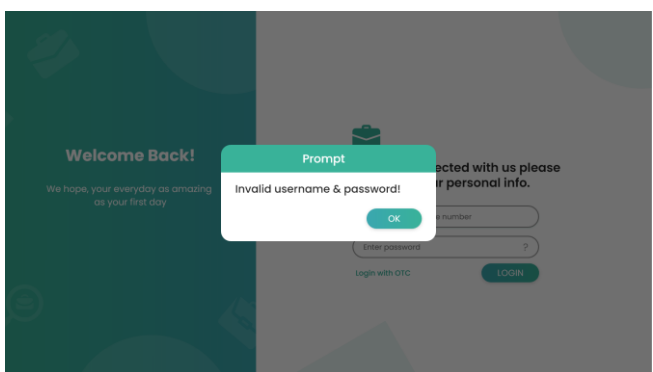
## 3. Decoy Files



Admin can view list of all the user's and can have a look on all the details and the activities of user. Moreover also can create & upload decoy files to make confusion for hacker to get into the system.

## Intruder Side:

### 1. Invalid User Account



If the hacker tries to hack the user account then after three attempts the hacker can get into fake account which has been created by admin which creates an illusion to hacker believing that he is using a user's real account.

## ACKNOWLEDGEMENT

This project report would not have been come into reality without the able guidance, support and wishes of all those who stand by us in the development. We wish to give our special thanks to our guide Prof. Rajashree Gadhave and Head of Department, Dr. Ashok Kanthe for their valuable guidance and unending support despite of a very busy work schedule. The cheerful spirit they radiated all the time fuelled our desire to excel in the work that we had undertaken. We acknowledge all the staff members of department of the Computer Engineering for their help and suggestions during various phases of this project work. It's difficult to forget our eminent supporters that are my family members and friends who are always their encouraging us in my every deed.

## REFERENCES

- [1] Ms. KomalNaik, Prof. VarshaBhosale and Prof. Vinayak D.Shinde, "Honeyword Generation Approach Using ASCII Values for Security Enhancement", March 2017.
- [2] Muhammad Ali Fauzi, Bian Yang, Edlira Martiri, "PassGAN-Based Honeywords System", Dec 2019.
- [3] Akshima, Donghoon Chang, Aarushi Goel, Sweta Mishra, Somitra Kumar Sanadhya, "Generation of Secure and Reliable Honeywords, Preventing False Detection", 2018.
- [4] SonaliBamane, Monika Shinde, Mrs.Amrपाली Mhaisgawali, Mangal Bargaje, Dr.Sanjay Pawar, "Achieving Flatness Using Honeywords Generation Algorithm", May 2019.
- [5] Neelam More, Minaj M. Pathan, Mahesh .B. Totre, Asst. Prof. Swati. S. Gore, "Honeyword: Achiving secure Passwords using HoneyEncryption", Nov 2016.

- [6] Omkar Madhavi, Avdhut Nalawade, Tejas Nagpure, Bharati Kavitate, "Intrusion Detection using Honeypots and Honeywords", March 2017.
- [7] A. Juels and R. L. Rivest, "Honeywords: Making Password-cracking Detectable," in Proceedings of the 2013 ACM.
- [8] J. Bonneau, "The science of guessing: Analyzing an anonymized corpus of 70 million passwords," in Security and Privacy (SP), 2012 IEEE Symposium on. IEEE, 2012, pp. 538-552.
- [9] Ding Wang, Haibo Cheng, Ping Wang, Jeff Yan, Xinyi Huang, "A Security Analysis of Honeywords", 2018.
- [10] Omar Z. Akif, G. J. Rodgers, G. J. Rodgers, H. S. Al-Raweshidy, "Achieving Flatness: Honeywords Generation Method for Passwords based on User Behaviours", March 2019.
- [11] Ajay Kumar, E. Jaya Krishna, "Privacy Enhancement For User Authentication Using Improved Honeyword Generation And Secure Hashing", March 2020.
- [12] Hackett, "Yahoo Raises Breach Estimate to Full 3 Billion Accounts, By Far Biggest Known," fortune.com, Oct. 3, 2017.
- [13] K. Akshaya and S. Dhanabal, "Achieving flatness from non-realistic honeywords," in 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), pp. 1-3. IEEE, 2017.
- [14] N. Chakraborty, S. Singh, and S. Mondal, "On Designing a Questionnaire Based Honeyword Generation Approach for Achieving Flatness," in 2018.
- [15] Z. A. Genc, S. Kardas, and K. M. Sabir, "Examination of a New Defense Mechanism: Honeywords," Cryptology ePrint Archive, Report 2013/696, 2013.