ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

Distributed Password Manager- Using Secure String Management technique

Syed Ali Mehdi

Assistant Professor, Department of Computer Science, Jamia Hamdard, Delhi, India

ABSTRACT: We tend to use more complicated passwords over the internet applications to ensure that no other person can access the application in use. There exists multiple applications where in we use different hence it becomes cumbersome for us to remember these passwords. Moreover we keep on changing the passwords frequently. Hence it becomes difficult for us to remember all the passwords. Many new applications have come in which take cares of all the passwords. Such applications manage passwords by storing the passwords in an encrypted form into their database. These applications are Distributed password managers. Many of these systems have their own set of advantages and disadvantages demerits motivate to go one step further. Hence in the proposed system to be developed we are using the concepts of Secure String Management technique and negative section algorithm of artificial immune system.

KEYWORDS: Artificial Immune system, Secure string management.

I. INTRODUCTION

Online services have changed the scenario of the world around us. Let it be online shopping, online money transactions, online communication, one needs to ensure the security. The simplest measure of security is the passwords. They help in identifying the authorized user. Over the recent years we have seen many network security attacks targeting passwords breaks so as to gain access to other accounts. Hence users keep on changing their passwords. Moreover a single user may have many supporting applications where in different passwords are being used. Like a user may have an account in Facebook, Gmail, PayPal, FTP and other applications with different user names and passwords. Thus it becomes cumbersome for the user to remember all the username and password used.

We tend to use more complicated passwords over the internet applications to ensure that no other person can access the application in use. There exists multiple applications where in we use different hence it becomes cumbersome for us to remember these passwords. Moreover we keep on changing the passwords frequently. Hence it becomes difficult for us to remember all the passwords. Many new applications have come in which take cares of all the passwords. Such applications mange passwords by storing the passwords in an encrypted form into their database. These applications are Distributed password managers. There are applications like password managers that help to ease such task. These online services database gets updated automatically. But most of them have their own advantages and disadvantages. Each of these applications has their own set of advantages and disadvantages. Some of them don't have best in GUI features like drag and drop option, some fail to warn if the selected password are not much strong. Some fail to maintain the history of passwords. These demerits motivate to go one step further. Hence in the proposed system to be developed we are using the concepts of Secure String Management technique and negative section algorithm of artificial immune system. The proposed work is to be developed in ASP.NET with My SQL as back end. The system is to be designed and evaluated. Hence in this research work a new application for distributed password management is being proposed developed based on the ideas of Secure String Management technique and negative section algorithm of artificial immune system.

II. RELATED WORK

Password managers

Password managers are acclimated to accomplish the passwords more secure. These applications use Password Hash techniques and Password Multiplier techniques. Password Hash techniques are used to generate strong passwords. These are browser plug-ins. Password Multipliers violates phishing attacks and are plug-ins.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

Password manager is a password protection vault. It is an application which is first installed in any device. It access username passwords and stores these details in to the database in encrypted form. This particular database is read only and is secured by special password. Hence one need to remember only one password and that is of password manager. There are a large number of open source, free, accessible passwords managers. Some key characteristics needed for a good password managers are:

- Use alone acclaimed and trusted solutions
- Must have auto updating of database.
- Must be having simple GUI features
- Must be executing on different platforms, like computer based on windows, Linux etc.
- Must have the ability to synchronize frequently with the device in use.
- Must warn for weak passwords selected.
- Must be secure

There are many such open source applications like keepass, password safe etc.

A. *Drawbacks for the existing system*

On examining them I find some demerits of the existing system.

- Some of them don't have best in GUI features like drag and drop option,
- Some fail to warn if the selected passwords are not much strong.
- Some fail to maintain the history of passwords.
- Some don't utilize features of cloud and AI.

B. *Artificial immune system*

Artificial immune system (AIS) models the natural immune system's ability to detect cells which are foreign to the body or are different from the body cells. The result is a new computational field with powerful pattern recognition abilities, mainly applied to anomaly detection[1].

Artificial immune systems can be characterized as unique or figurative computational frameworks created utilizing thoughts, speculations, and parts, of human immune system. Most AIS go for taking care of complex computational or building issues, for example, design acknowledgment, end, and optimisation[2].

C. *Biological and Artificial Immune Systems*

All living organisms are capable of presenting some type of defense against foreign attack. The evolution of species that resulted in the emergence of the vertebrates also led to the evolution of the immune system of this species.

The immune system of vertebrates is composed of a great variety of molecules, cells, and organs spread all over the body. There is no central organ controlling the functioning of the immune system, and there are several elements in transit and in different compartments performing complementary roles. The main task of the immune system is to survey the organism in the search for malfunctioning cells from their own body (e.g., cancer and tumour cells), and foreign disease causing elements (e.g., viruses and bacteria). Every element that can be recognised by the immune system is called an *antigen* (Ag).

The cells that initially fit in with our body and are harmless to its working are termed self (or self antigens), while the infection bringing about components are named nonself (or nonself antigens). The immune framework, subsequently, must be fit for recognizing what is self from what is nonself; a methodology called self/nonself separation, and performed fundamentally through example recognition occasions. From an example recognition point of view, the most engaging normal for the will be the vicinity of receptor atoms, on the surface of safe cells, fit for perceiving a just about boundless scope of antigenic examples. One can distinguish two noteworthy gatherings of resistant cells, known as B-cells and T-cells. These two sorts of cells are somewhat comparative, however vary with connection to how they

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

perceive antigens and by their utilitarian parts. B-cells are equipped for perceiving antigens free in arrangement (e.g., in the circulatory system), while T-cells oblige antigens to be displayed by other extra cells. Antigens are secured with atoms, named epitopes. These permit them to be perceived by the receptor particles on the surface of B-cells, called antibodies (Ab).

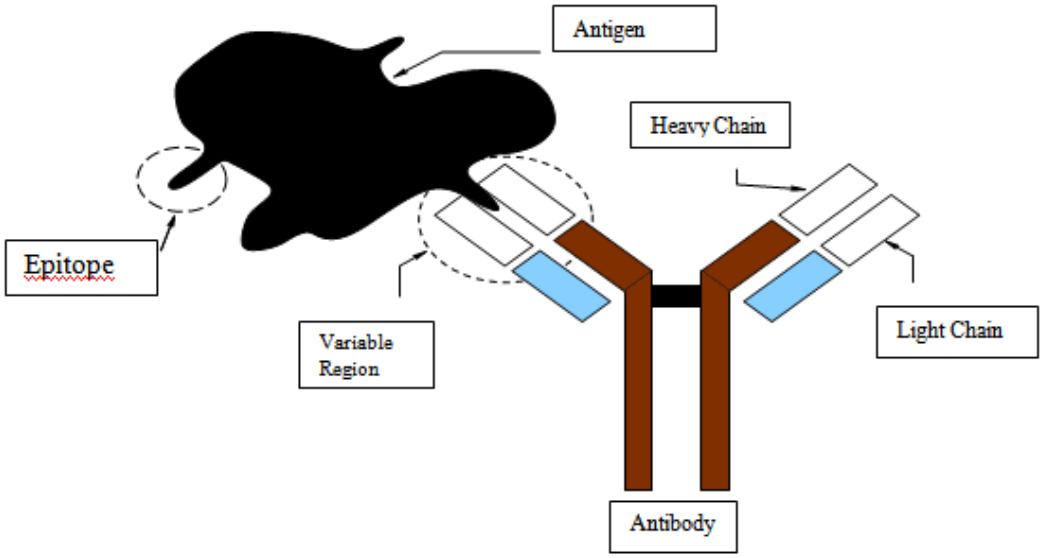

Fig 1.Antibody antigen complexes. The diagram explains the way the antigen and the antibody forms a bond.

D. ***AIS algorithms***

This section presents definitions of the key terms and a concept used throughout the rest of this work and explains the basic artificial immune system algorithm.

- Affinity: a measure of "closeness" or similarity between two antibodies or antigens. In the current implementation, this value is guaranteed to be between 0 and 1 inclusively and is calculated simply as the Euclidean distance of the two objects' feature vectors. Thus, small affinity values indicate strong affinity.
- Affinity threshold (AT): the average affinity value among all of the antigens in the training set or among a elected subset of these training antigens.
- Affinity threshold scalar (ATS): a value between 0 and 1 that, when multiplied by the affinity threshold, provides a cut-off value for memory cell replacement in the AIRS training routine.
- Antibody: a feature vector coupled with its associated output or class; the feature vector-output combination is referred to as an antibody when it is part of an ARB or memory cell.
- Antigen: It is similar to the antibody but the feature vector class becomes an antigen when it is given to the ARBS for any response.
- Artificial Immune Recognition System (AIRS): a classification algorithm inspired by natural immune systems.

It is mainly the inner working and cooperation between the mature T-Cells and B-Cells that is responsible for the secretion of antibodies as an immune response to antigens. The T-Cell becomes mature in the thymus. A mature T-Cell is self-tolerant, i.e. the T-Cell does not bind to self-cells. The mature T-Cell's ability to discriminate between self-cells and non-self-cells make the NIS capable of detecting non-self-cells. The affinity can be seen as a measurement to the number of lymphocytes that must be secreted by the T-Cell to clonally proliferate the B-Cell into a plasma cell that can produce antibodies [1] The artificial lymphocytes are represented as ALC.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

Algorithm1 Basic AIS Algorithm

C is initialized with a set of ALC.

DT be the antigen training patterns

While *some stopping condition(s) is not true* do
  for each *antigen pattern* $A_p \in D_T$ do
    Select a subset of ALCs for exposure to A_p,as population $S \subseteq C$;
    for each *ALC,* $x_i \in S$ do
      Compute the *antigen affinity between* A_p and x_i;
    end
    Select a subset of ALCs with the highest calculated *antigen affinity as* population $H \subseteq S$;
    Adapt the ALCs in *H* with some *selection method*, based on the calculated
    antigen affinity and/other *network affinity* among ALCs in *H*;
    Update the *stimulation level* of each ALC in *H*;
  end
end

In initialization, data pre-processing is carried out with a parameter discovery stage. All items in the data set are normalized where the Euclidean distance between any two feature vectors is [0, 1].
After normalization, the affinity threshold is calculated. Then the training proceeds. Memory cell with highest affinity is identified and steps are repeated till condition is not satisfied. The ALC with highest affinity is selected.

E. ***Negative selection algorithm***

One of the AIS models based on the classical view of the natural immune system is the model introduced by Forrest *et al[2]*. In this classical AIS ,Forrest *et al.* introduced a training technique known as *negative selection*. In the model, all patterns and ALCs are represented by nominal- valued attributes or as binary strings.
The affinity between an ALC and a pattern is measured using the *r*-continuous matching rule. The *r*-continuous matching rule is a partial matching rule, i.e. an ALC detects a pattern if there are *r*-continuous or more matches in the corresponding positions. *r* is the degree of affinity for an ALC to detect a pattern. A higher value of *r* indicates a stronger affinity between an ALC and a pattern.[1]
For a problem first define the patterns which are to be protected as the Self-Set(P). It is now to generate detector set (M) which shall identify the elements who don't belong to the self- set. Thus the negative selection algorithm runs as:
1. First a random element C is generated
2. then comparison is carried out between elements of C and P. in case a match occurs i.e. elements of P are able to recognize the elements of C then those elements of C are discarded otherwise the elements of C are stored in the detector set M.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

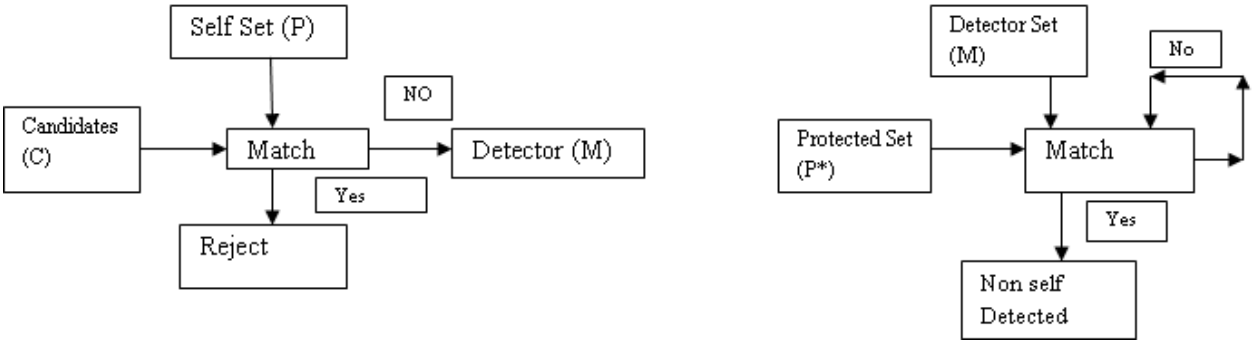

Fig 2.The diagram explains the way in which matching of negative selection algorithm is carried out. Generating Detector set and matching using Negative selection algorithm.

Now when set of M is generated the next step is to monitor the system for finding the non-self patterns. Assuming P* as the patterns that are to be protected which may comprise of set of P and other new patterns. The next step is to find a match of elements of P* and the detector set M. If a match is found then that nonself pattern is recognized. Thus a nonself pattern is detected.

Algorithm2 Negative Selection Algorithm

Set counter n_a as the number of self-tolerant ALCs to train; Create an empty set of self-tolerant ALCs as C;
Determine the training set of self patterns as D_T;
while *size of* C *not equal to* n_a do
Randomly generate an ALC, x_i;
Matched=false;
for *each self pattern* $z_p \in D_T$ do
if *affinity between* x_i *and* z_p *is higher than affinity threshold r* then
matched =true;
break;
end
end
if *not matched* then
Add x_i to set C;
end
end

III. PROPOSED METHOD

The proposed system is designed using the concept of secure string management technique and negative selection algorithm of artificial immune system. The system addresses these issues:

- To design a light weighted application with easy to use graphical user interface so as to manage the passwords in secured form.
- To manage your entire password for different websites with username and also the website address to which the password belongs to.
- In case if any user forgets the password then the application shall provide an option to view the saved password.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

- To create multiple books to save details of different persons.

Algorithm 3 : for proposed system

Start
Step 1:Login to the distributed password manager by registering in to the application.
Step 2: using drag and drop options customize your account
Step3: list the user names and passwords along with the websites to which they belong to.
Step 4: all the passwords shall undergo encryption using proposed encryption technique based on artificial immune system.
Step 5: store the passwords in the database
Step6: using secure string management concept store the file of the user in to the associated cloud of the database. This enables the application to be used online.
Stop

Algorithm 4 : for encryption and decryption

Step 1: Take password as the input message.
Step 2: select a random password as a key. This shall be the fitness criteria. To select this password for a set of random passwords, calculate the similarity or closeness between the supposed password and elements of the stored passwords. If the supposed password does not recognize any body then it goes to the set of random passwords.
Step3: This selected password is the key and all passwords and data are encrypted using this selected password. Encryption of password = (selected password (key), password).
Step 4: Reverse is carried out for decryption.
Stop.

IV. EXPERIMENTAL RESULTS

The proposed design for password manager was successfully designed using ASP. NET and MY SQL. It has very user friendly GUI and supports safety of stored passwords.

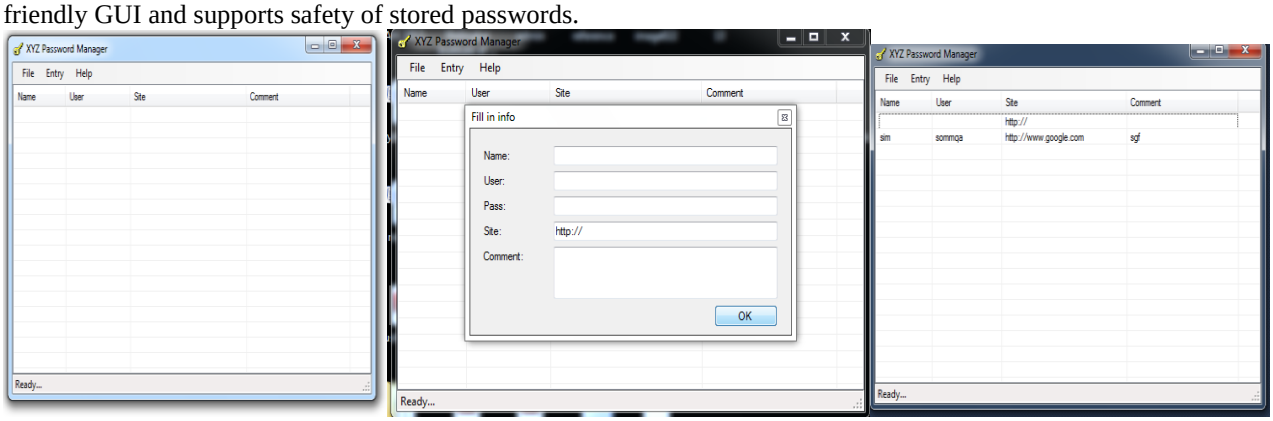

Fig 3.The Experiment result where password is stored successfully. This Figure explains how a sample password string was stored and managed easily. After successful saving the content are displayed as above.

V. CONCLUSION

The idea of storing passwords safely is very useful for persons or organization wherein they have multiple accounts. To ensure its safety they opt password manager applications. In this work a new distributed password manager was designed and implemented in ASP. NET and MY SQL using the concept of secure string management and encryption

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

based on artificial immune system. The immune system tool box was used for the said purpose. The designed application is user friendly and does warn if the chosen password is weak.

VI. REFERENCES

[1] Andries P. Engelbrecht. Computational Intelligence-An Introduction Second Edition- University of Pretoria South Africa.
[2] L. N. de Castro and J. Timmis. Artificial Immune Systems:A Novel Paradigm to Pattern Recognition In Artificial Neural Networks in Pattern Recognition , J. M. Corchado, L. Alonso, and C. Fyfe (eds.), SOCO-2002, University of Paisley, UK, pp. 67-84, 2002.
[3] Chandra, S., K.R. Sudha and P.V.G.D. Prasad Reddy,. Data Encryption technique using Random number generator. IEEE International Conference on Granular Computing, 2007,pp: 576-579.
[4] Chen, S. and X.X. Zhong, Chaotic block iterating method for pseudo-random sequence generator. J. China Univ. Posts Telecommun.,2007. 14(1): 45-48.
[5] Forrest, S. and A.S. Perelson, 1994. Self-Nonself Discrimination in a Computer [J]. IEEE Symposium on Security and Privacy. 1994. IEEE Computer Society Press, Oakland, CA, pp: 202-213.
[6] Forrest, S., S. Hofmeyr and A. Somayaji, Computer immunology [J]. CACM,1997, 40(10): 88-96.
[7] Huang, G. and X. Li, An intrusion detection system based on immune principle [J]. Micro Electr. Comput.,2008. 25(8): 192-194, (in Chinese).
[8] Han K H, Park K H. Parallel Quantum-inspired Genetic Algorithm for Combinatorial Optimization Problems, Proceedings of the CEC. Piscataway: IEEE Press, 2001:1442-1429.
[9] Retrieved from http://www.purdue.edu/securepurdue/pswdManager.cfm