International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 3 Issue: 11 6175 - 6184

Hybrid Security Framework for Activity Based Authentication using RSA & Genetic Algorithm

Mr. P.M.More
Research Student
Chh Shahu Institute of Business Education and Research, Kolhapur, India
pandumore.1603@rediffmail.com

Dr. Poornima G. Naik
Professor, Department of Computer Studies
Chh Shahu Institute of Business Education and Research, line Kolhapur, India
luckysankalp@yahoo.co.in

***Abstract*—** In the current information age, security has achieved a tremendous importance in e-commerce applications involving financial transactions. Non-repudiation, data integrity, data confidentiality and authenticity, have become an integral part of information security. There is a tremendous risk involved in the communication of a plain text over Internet. Cryptography offers a solution for this type of risk which is referred to as a technique of encrypting and decrypting messages in such a way that they cannot be interpreted by anybody with the exception of a sender and an intended recipient. In majority of the e-commerce based applications where security is considered to be of prime importance, a single encryption algorithm is adopted for encrypting a password and the authentication information is stored on a single database server which becomes open to risks against different computer hacks. A novel solution for this problem is to generate an individual's personal and dynamic activities which will be hard for the attackers to guess. Further, this can be combined with distributed technology where the authentication information is distributed over geographically separated multiple servers. In this paper authors have generated an activity based distributed 3D password incorporating various activities where the authentication information is distributed over geographically separated multiple authentication servers. The key pair is generated using RSA algorithm which is encrypted using single-point cross over and mutation of bits at the extreme position. This further adds another level of security and renders the key unbreakable by an unintended user. The configuration information pertaining to the distributed environment is stored in XML file which is parsed using Microsoft's XML Parser and the activity related information is stored in different servers which is encrypted using RSA algorithm. The technique employed combines RSA algorithm with Genetic Algorithm to offer a robust hybrid security framework in a distributed environment which is difficult to guess for an unintended user .

Keywords- *Activity, Authentication, Distributed Password, Genetic Algorithm, RSA Algorithm, Winsock.*

*****

I. INTRODUCTION

In any organization it is an extremely essential task to protect the data from unauthorized users. Information systems hardware, software, networks, and data resources need to be protected and secured to ensure quality, performance, and reliability. Security management is the accuracy, integrity, and safety of information resources. When effective security measures are in place, they can reduce errors, fraud, and losses. The Internet increases the vulnerability of information systems and networks so that they can be used to facilitate attacks by criminals, unauthorized users and hackers. There are different kinds of attacks such as brute force attacks, dictionary attacks etc to name a few for stealing and misusing passwords for gaining an unauthorized access to the system. So the researchers in this area are continuously focusing on design of most robust security frameworks which are highly secure and difficult to break. One such framework deals with activity based authentication and 3D distributed password. But as the literature reveals, such a framework is based on crisp logic which makes the system less reliable. However, if the fuzziness is introduced in the existing security framework the reliability can increase many fold. Further, another layer of confidentiality and reliability can be offered by distributed environment where the encryption key is broken into n-sub keys and is stored in a framework of geographically distributed authentication servers.

In traditional organizations the system is less secure where some unauthorized users, hackers can be misuse of organizations private data. The traditional system has many disadvantages since it is associated with high administrative that has high possibilities of misuse of organizations important data with very low data security.

The activity based authentication of distributed environment of any organization has definitely an upper hand as compared to all other authentication techniques. The approach comprises of authentication through the use of an individual's personal and dynamic Internet activities. The authors hypothesize that frequently-changing secret question which are based on time of a day, or day of a week, will be hard for attackers to guess. Activity based authentication can be achieved by neural networks, fuzzy logic etc. This security tool authenticates users at the operating system level in multi-user operating systems. It supports system administrators in limiting the ability of unauthorized users to disrupt system operations using a neural network and set of rules to track usage patterns, color code change and object transition activity on the system.

The aim of research is to study various soft computing techniques such as fuzzy logic, neural network and genetic algorithm which can be applied for activity based authentication in a distributed environment to protect data& transactions against unauthorized users.

Objectives and Significance of Research

The main objective of study is to provide security of organization by using latest soft computing techniques-

- To provide high level security mechanism to the organization.
- To protect organization from harmful effect from unauthorized users and transactions.
- To build highly secure organization.
- To implement a tool which can be employed by any organization based on the security level desirable by the organization.
- To store a data in an encrypted form where the encryption key is divided into sub keys and stored on different geographically distributed authentication servers.

The distributed application designed with the help of soft computing techniques will help to provide high level security within particular organization. Another layer of security is added by a distributed environment. The tool can be employed by any organization interested in protecting its confidential data. The tool addresses all the issues pertaining to security from low security level to high security level. It protects information& transaction within organization against unauthorized users and it is only accessible to authorized users due to that approach information remains secure. As authentication data is dynamic, time dependent and is frequently changing, it is extremely difficult for an intruder to gain an unauthorized access to the system. Soft computing techniques can increase accuracy, integrity, and safety of information resources.

- To manage security in organization
- To give access to only authorized persons and preventing from unauthorized

II. LITERATURE REVIEW

One of the important aspects of security system is authentication. Authentication is something what user knows, what user is or what user has. It could be a unique text phrase known only to the user, or could be physical tokens such as bar code or RFID card which user possesses or it could be his physical features such as facial features, thumb impressions unique to that user. There are many attempts in designing fool proof authentication systems which are robust and difficult to break. The most robust being biometric authentication, which has its own positive and negative sides inherent into it. Since they are hard to refute, effective utilization of any of these techniques can effectively identify a person [1,5] with minimum errors. Different types of authentication mechanisms provide authentication mechanisms of different levels of complexity. In the literature there are handful of authentication mechanisms in place which focus on activity based personal questions, incorporating network activities, physical events, conceptual questions etc.[6,7]. In their paper authors [8] have proposed an activity-based authentication framework and have described their preliminary evaluation results on its usability and security. Their approach is focused towards improving the robustness of existing question-based authentication systems emphasizing on the use of short-lived questions that are automatically extracted from the underlying database storing user's personal Internet activities which are hard to predict. Activity-aware ECG-based patient authentication system for remote health monitoring was carried out by Sriram et.al. [9]. In their paper , they have presented a novel ECG and accelerometer-based system that can authenticate individuals in an ongoing manner under different activity conditions. They have presented the probabilistic authentication system and have presented experimental results from 17 individuals. A review on the challenges that are encountered in Biometric based authentication techniques was carried out by Sheela Shankar et.al. [10]. In their paper they have carried out a survey of the challenges encountered in two biometric based approaches: face recognition technique and authentication via neuro-signals. According to their survey, it is observed that the key challenges in the former technique are less complex when compared to the latter technique.

III. CONCEPTUAL MODEL

The structure of the XML file for storing activity information in a distributed environment is as shown below:

```
<activities>
    <activity>
            <name>TimeActivity</name>
            <machineName>localhost</machineName>
            <port>5000</port>
    </activity>
    <activity>
            <name>ImageActivity</name>
            <machineName>localhost</machineName>
            <port>6000</port>
    </activity>
<activity>
            <name>PatternActivity</name>
            <machineName>localhost</machineName>
            <port>7000</port>
    </activity>
    <activity>
            <name>MixColorActivity</name>
            <machineName>localhost</machineName>
            <port>8000</port>
    </activity>
</activities>
```

The corresponding Document Type Definition (DTD) is as shown below:

```
<?XML version="1.0" ?>

<!ELEMENT activities(activity+)>

<!ELEMENT activity(name, machineName, port)>

<!ELEMENT name(#PCDATA)>

<!ELEMENT machineName(#PCDATA)>

<!ELEMENT port(#PCDATA)>
```

Proposed Algorithm

Encryption Algorithm :

Let K_1 and K_2 represent the public and private keys generated by RSA algorithm, respectively. Let $L(K_1)$ and $R(K_1)$ represent left and right blocks containing 4-bits each of key K_1. Similarly, $L(K_2)$ and $R(K_2)$ represent left and right blocks containing 4-bits each of key K_2.

Hence, $K_1 = L(K_1) + R(K_1)$

$K_2 = L(K_2) + R(K_2)$

After performing cross-over operation at the single cross-over point selected at the mid point the new keys generated are

$K_1 = R(K_2) + R(K_1)$

$K_2 = L(K_2) + L(K_1)$

On performing mutation operation at the extreme positions by complimenting the bits, the new keys generated are,

$K_1 = mut(K_1)$

$= mut(R(K_2)) + mut(R(K_1))$

and

$K_2 = mut(K_2)$

$= mut(L(K_2)) + mut(L(K_1))$

where, mut() is a mutation operation which in our case, compliments the bits at the extreme position.

Decimal representation of K_1 and K_2 represent the encrypted public key and private key, respectively.

Reverse steps are employed for decrypting the key pair.

Let E(PublicKey) and E(PrivateKey) represent encrypted public key and private key, respectively, where

E(PublicKey) = mut(cross_over(PublicKey)).

mut(), cross_over() represent mutation and single point cross-over operations which are symmetric in nature. Hence exploiting the symmetry of these operations, we have

PublicKey =cross_over(mut(E(PublicKey))) (1)

Similarly,

PrivateKey =cross_over(mut(E(PrivateKey))) (2)

Let M represent the activity message to be encrypted.

Encryption of the message using public key using RSA algorithm is given by,

$E^{RSA}_{PublicKey}(M)$

Decryption of the message using private key using RSA algorithm is given by,

$D^{RSA}_{PrivateKey}\{ E^{RSA}_{PublicKey}(M) \} \equiv M$

where the public and private keys are retrieved from their encrypted counter parts employing the operations depicted in eq^n(1) and eq^n(2), respectively.

Application of Encryption/Decryption Algorithm to a key pair.

Encryption Algorithm

Let the public key and private key generated by RSA algorithm be 17 and 29, respectively.

Step 1 : Encode the public key and private key into 8-bit code containing left and right blocks of 4-bits each.

			$L(k_1)$				$R(k_1)$
0	0	0	1	0	0	0	1
			$L(k_2)$				$R(k_2)$
0	0	0	1	0	1	0	1

Step 2 : Perform cross-over operation at the single cross-over point selected at the mid point.

			$R(k_2)$				$R(k_1)$
1	1	0	1	0	0	0	1
			$L(k_2)$				$L(k_1)$
0	0	0	1	0	0	0	1

Step 3 : Perform mutation operation at the extreme positions by complimenting the bits.

			$R(k_2)$				$R(k_1)$
0	1	0	1	0	0	0	0
			$L(k_2)$				$L(k_1)$
1	0	0	1	0	0	0	0

Step 4 : Decode the public key and private key into decimal numbers to obtain the encrypted public and private keys, respectively.

Encrypted Public Key : 80 Encrypted Private Key : 144.

Decryption Algorithm

Step 1 : Encode the encrypted public key and private key into 8-bit code containing left and right blocks of 4-bits each.

			$R(k_2)$				$R(k_1)$
0	1	0	1	0	0	0	0
			$L(k_2)$				$L(k_1)$
1	0	0	1	0	0	0	0

Step 2 : Perform reverse mutation operation at the extreme positions by complimenting the bits.

			$L(k_1)$				$R(k_1)$
1	1	0	1	0	0	0	1
			$L(k_2)$				$R(k_2)$
0	0	0	1	0	0	0	1

Step 3 : Perform cross-over operation at the single cross-over point selected at the mid point.

			$L(k_1)$				$R(k_1)$
0	0	0	1	0	0	0	1
			$L(k_2)$				$R(k_2)$
0	0	0	1	1	1	0	1

Step 4 : Decode the public key and private key into decimal numbers to obtain the encrypted public and private keys, respectively.

Decrypted Public Key : 17 Encrypted Private Key : 29

The partial Java program for generating public key/private key pair is given in Appendix A.

Application Architecture

The framework for storing the activity-based information in a MS-Access database is depicted in Figure 1. The activity related information accepted from an end user is encrypted using RSA algorithm by retrieving the GA-encrypted public key from the key store database which is decrypted using Genetic Algorithm. The decrypted public key so generated is then employed in encryption using RSA algorithm. The resulting encrypted data is stored in a distributed server as configured by an end user.

Storing Activity Information in a Database

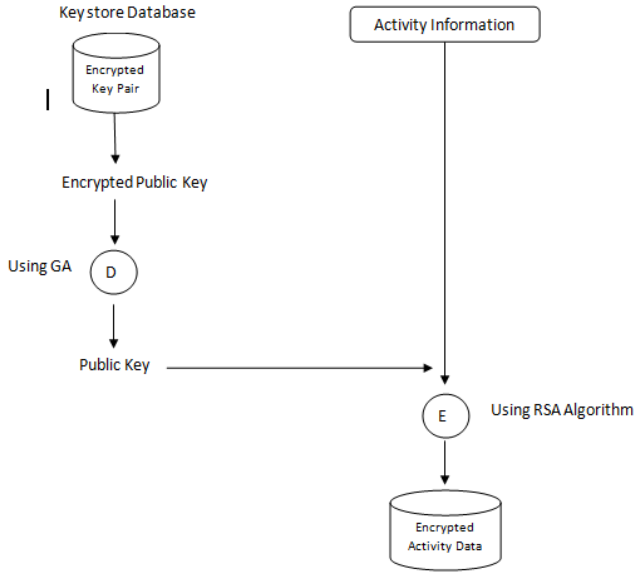


Figure 1. Framework for storing the activity-based information in a MS-Access database.

Activity based Authentication Process.

Activity-based authentication process is characterized into two distinct models, Type-I and Type-II which are shown in Figure 2 and Figure 3, respectively. In Type-I authentication model, encrypted activity information stored in the database is decrypted using RSA algorithm using corresponding private key which is then compared with activity-based information entered by an end user. In Type-II authentication model, activity-based information entered by an end user is encrypted using the RSA algorithm using public key stored in key store database which is then compared with encrypted activity-based information present in the database.

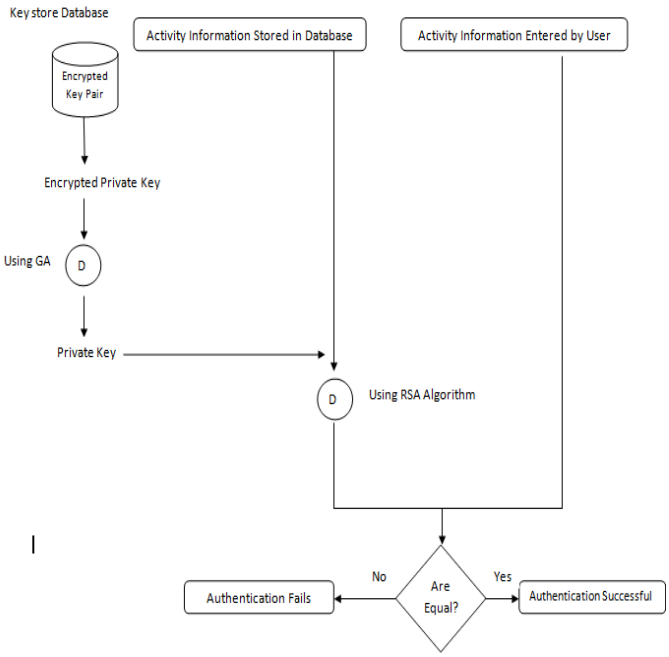


Figure 2. Type-I Authentication Process Model

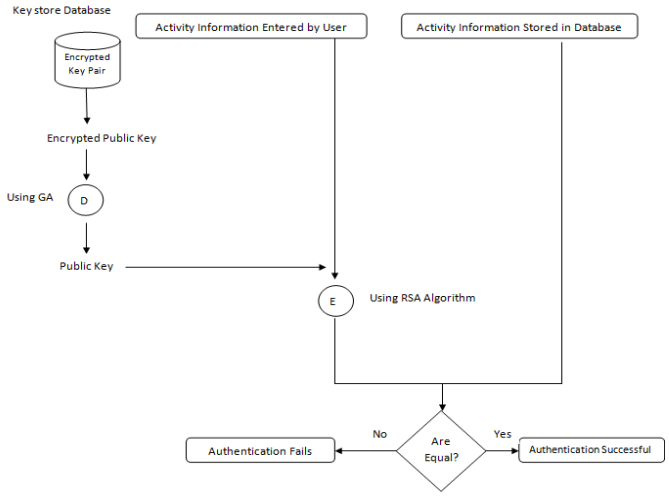


Figure 3. Type-II Authentication Process Model.

Type-II model is more robust as compared to Type-I model as encrypted data is transmitted over the network medium in contrast to Type-I model where plain data is transmitted if authentication occurs on server-side.

The sequence of steps occurring at both the client and the server during the storage of encrypted activity data in a database and authentication process is shown in Figure 4.

Generation of Public Key Private Key Pair	
Encryption of key pair using Genetic Algorithm	
Storing encrypted keys in a key store database.	Comparing generated Activity information with the information entered by an end user.
Reading encrypted public key from a key store database.	Decryption of Activity Data using Private Key using RSA Algorithm
Decryption of public key	Decryption of private key
Encryption of Activity Data using Public Key using RSA Algorithm	Reading encrypted private key from a key store database
Storing encrypted Activity Data in a Database	Retrieving encrypted Activity Data from a Database

Figure 4. Sequence of Steps during Application's Execution.

 ISSN: 2321-8169
6175 - 6184

A layered architecture along with the function of each layer is depicted in Figure 5.

Presentation Tier (VB) End User Interaction and Distributed Environment Management
Middle Tier (Java) Encryption and Decryption of Data
Data Tier (MS - Access) Storage of Retrieval of Data

Figure 5. Layered Application Architecture.

Control Flow Diagram

The control flow diagram fro key generation and encryption using Genetic Algorithm is shown in Figure 6.

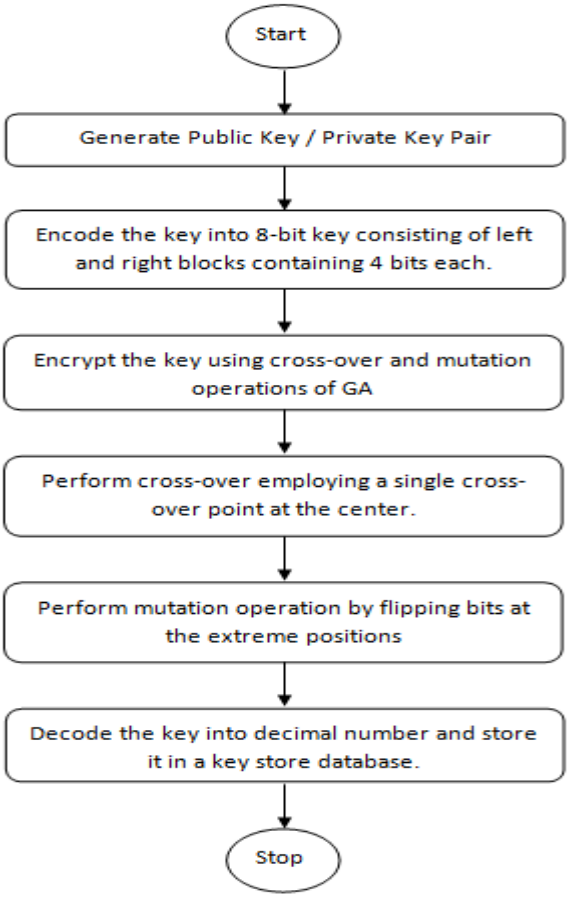


Figure 6. Key Generation and Encryption using Genetic Algorithm

Control flow diagram for parsing XML File containing Distributed Server Configuration Information is shown in Figure 7.

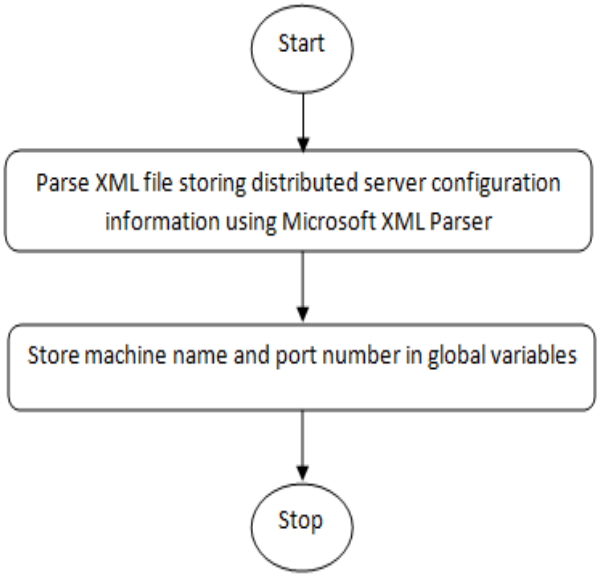


Figure 7. Parsing XML File containing Distributed Server Configuration Information

Control flow diagram for Storing Activity based information in a Distributed Authentication Server is shown in Figure 8.

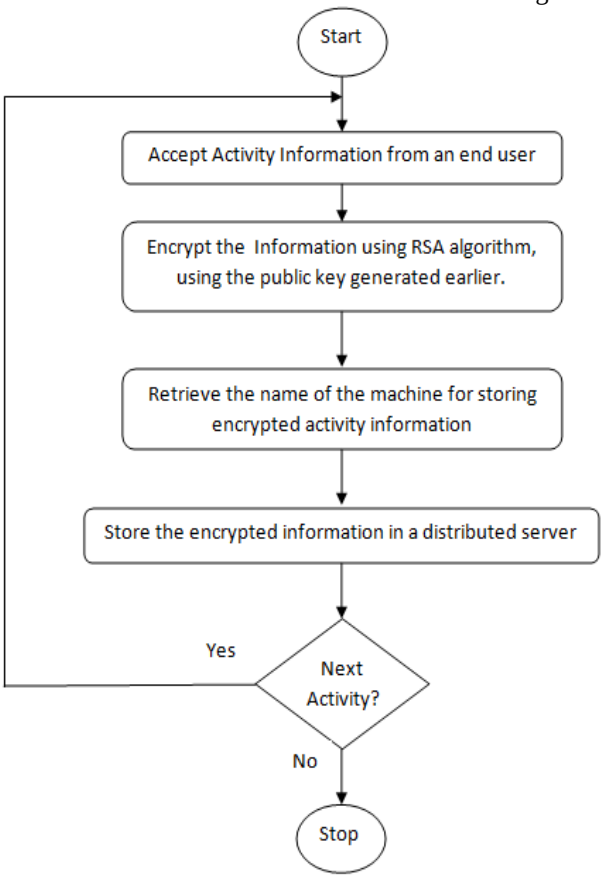


Figure 8. Storing Activity based information in a Distributed Authentication Server

The control flow diagram for activity-based authentication process is shown in Figure 9.

ISSN: 2321-8169
6175 - 6184

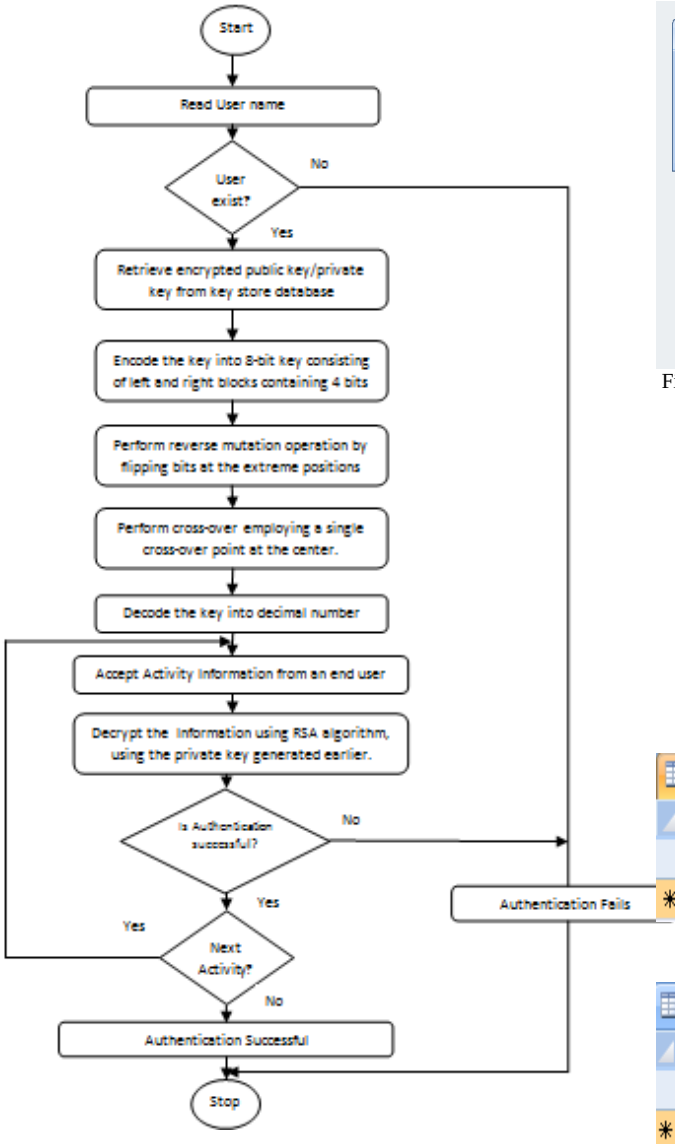


Figure 9. Activity-Based Authentication Process.

IV. RESULTS AND DISCUSSIONS

The model for activity authentication tool is implemented in VB with MS-Access as backend for storing authentication related information. The structure of the database is shown in the following Figure 10. Figure 11 shows key pair encrypted using Genetic Algorithm which is stored in a key store database. Figures 12(a)-12(d) show encrypted activity-based information stored in MS-Access database. Winsock control is used for remote connection. The authentication information is distributed on multiple servers by employing vertical fragmentation. The user can implement the database on multiple machines and select the machine for both storing the authentication information and for performing end user authentication. The user interface for the start up application screen and connection to the remote server using winsock control are depicted in Figure 13 and Figure 14, respectively.

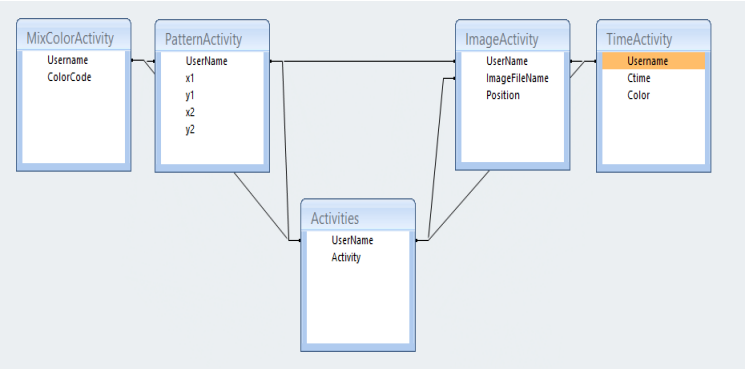


Figure 10. Structure of Database for Storing Activity-based Authentication Information.

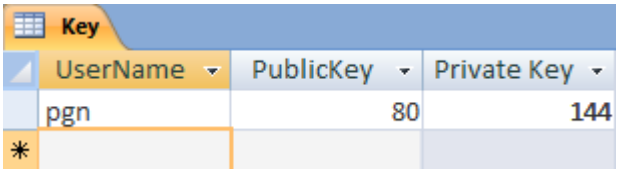


Figure 11. Encrypted Key Pair Stored in Key Store Database.

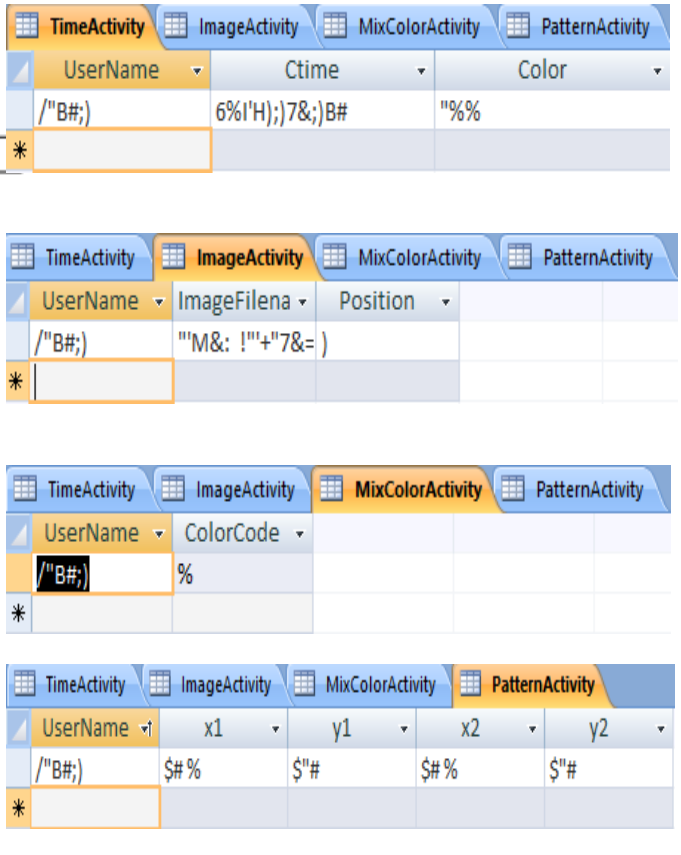


Figure 12 (a) – 12 (d). Sample Encrypted Activity Based Information Stored in MS-Access Database.

 ISSN: 2321-8169
6175 - 6184

Figure 13. GUI for Startup Application Screen

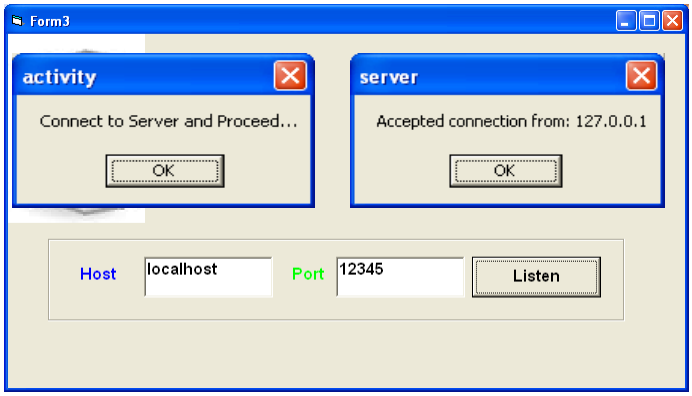


Figure 14. GUI for Connecting to Remote Server using Winsock Control.

Figures 15 (a)- 15 (d) depict generating and storing information pertaining to the following activities.

- Time Activity
- Image Activity
- Pattern Activity
- Mix Color Activity

In Time Activity, the user can select three different colors corresponding to three different times of the day where the time zones are created in 24-Hour format as shown below:

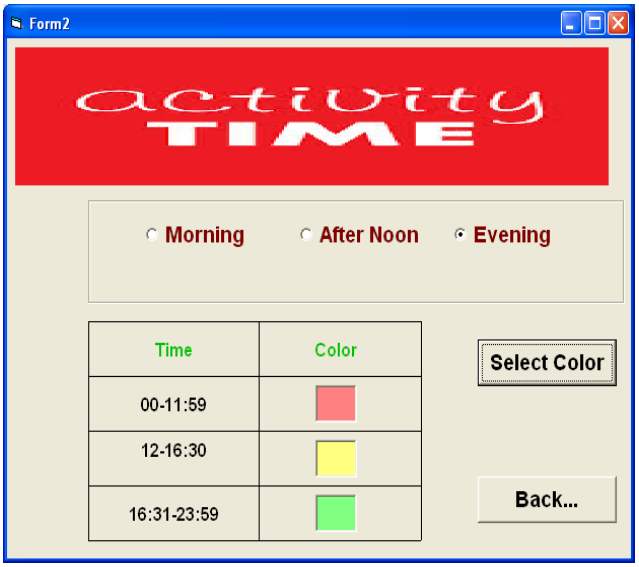
Morning	0-11.59
Afternoon	12-16.30
Evening	16-31-23.59

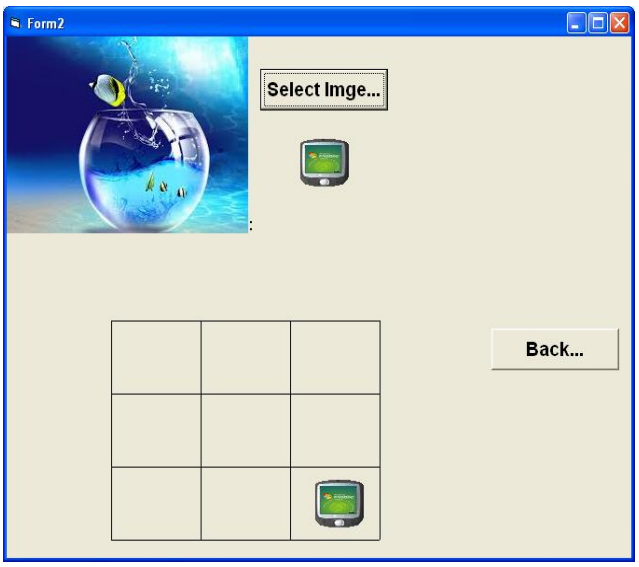
The tool will authenticate the user appropriately by selecting the color based on the time of the day.

In the image activity, the user can select and move the image from the upper-left grid to any of the grid locations of 3x3 matrix which is recorded by the tool and the user is authenticated accordingly. The pattern activity comprises of generating a pattern by joining the tiny solid circles in any fashion. The final mix color activity enables the end user generating any one of the following colors by mixing zero or more primary colors as depicted in the following table.

R	G	B	Resultant Color
x	x	x	Black
✓	x	x	Red
x	✓	x	Green
x	x	✓	Blue
✓	✓	x	Yellow
✓	x	✓	Magenta
x	✓	✓	Cyan
✓	✓	✓	White

Figures 16(a)-16(d) show GUI for activity-based authentication process.





 ISSN: 2321-8169
6175 - 6184

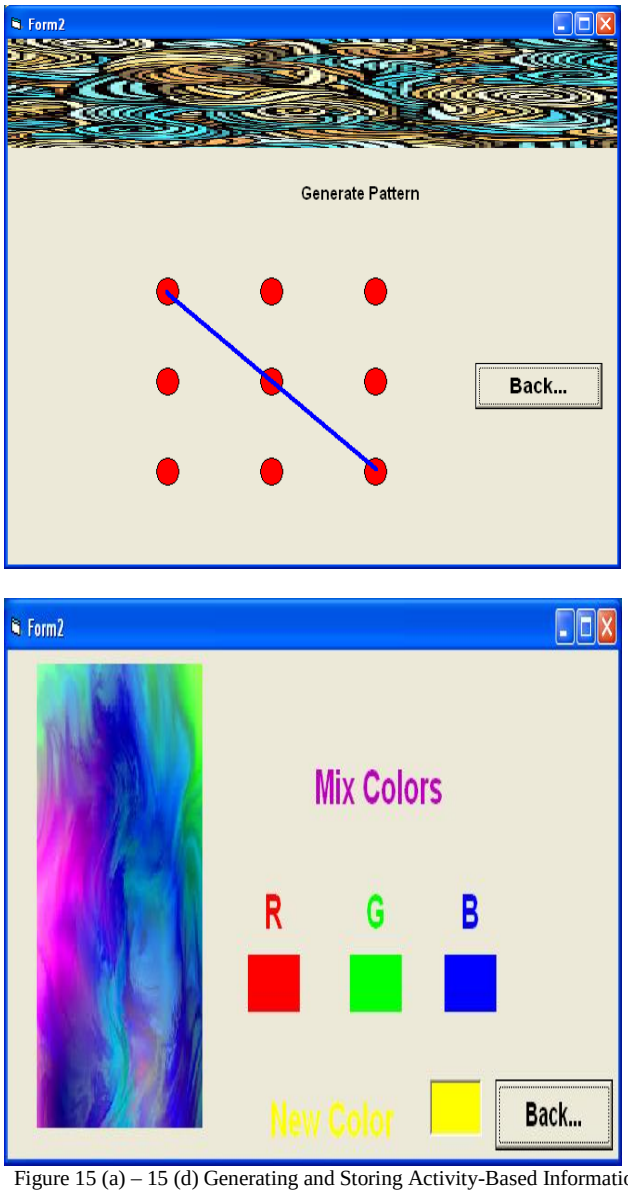
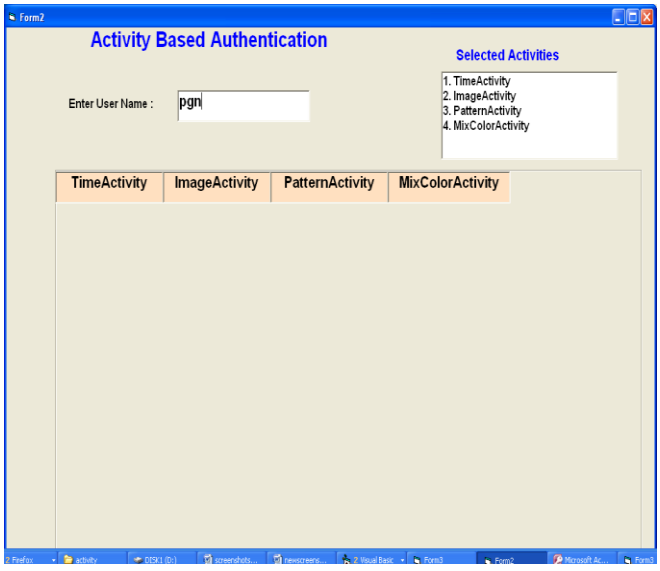


Figure 15 (a) – 15 (d) Generating and Storing Activity-Based Information in a Database.



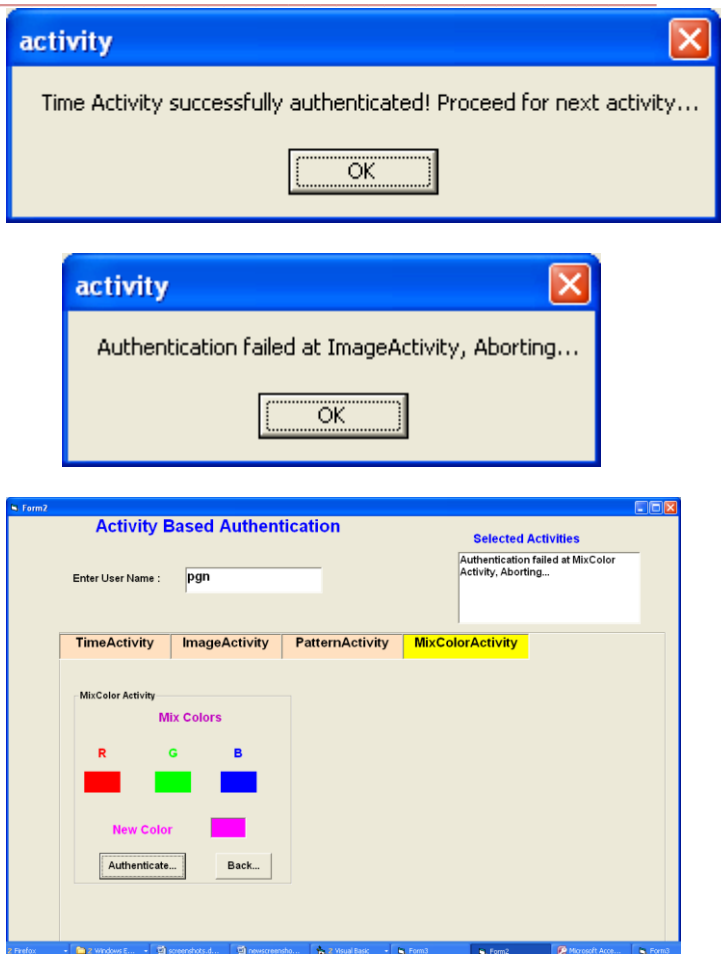


Figure 16 (a) – 16 (d) GUI for Activity-Based Authentication Process.

V. Conclusions and Scope for Future Work

In the current work, the authors have made an attempt to generate an activity based distributed 3D password incorporating various activities where the authentication data is stored over multiple distributed authentication servers. The public key/ private key is generated and encrypted using single-point cross-over and mutation operations of Genetic Algorithm and stored in a key store database. The client uses winsock control for communication with the server. Currently, four different activities are considered, however the activity set can be extended further as new activities are coined and added to the set constantly. In future we plan to adopt two different architectures using RMI and encryption using Neural Network employing two-tier architecture and a three-tier architecture. In a two-tier architecture, RMI server and RMI client reside on two different physical as well as logical tiers. RMI server is tightly integrated with the data tier. The application can be designed to be more manageable by switching to 3-tier application architecture by separating out database server from the RMI server and further using RMI security manager for enabling remote connection. Based on the level of security required and infrastructure availability in the organization, the end user can make a selection

between 1 to 4 different levels of security and authentication methods to be employed. What will be common to all these methods is that all of them will adopt Genetic Algorithm as a common technique for encryption and decryption of the key pair employed for encryption of password components wherein we will exploit the pseudo randomness involved in crossover and mutation operations for generating an asymmetric key pair for the encryption and the decryption of a password. The number of mutation points and number of crossover points together state the length of the secret key and hence measure the strength of the algorithm employed The authentication data will be encrypted on the client side and such encrypted data will be transmitted over the network which will then be compared with the encrypted information stored on respective authentication servers. Both during transmission of data and its storage, data will be encrypted in order to incorporate the data confidentiality tightly in to the application. The randomness along with the permutation process makes the algorithm robust and also hard to break. Further, the fitness function of GA can be designed to measure the strength of security. The algorithm in implemented in Java and applied for the encryption and decryption of an activity-based password of employees in an hypothetical organization.

REFERENCES

[1] B. Miller, Vital signs of identity, IEEE Spectrum 31 (2) (1994) 22–30.

[2] Jain, A. K., Ross, A., , & Prabhakar, S. 2004. An introduction to biometric recognition. IEEE Transactions On Circuits and Systems for Video Technology, 14, 4–20.

[3] Enrique G. Ortiz, Brian C. Becker, “Face recognition for web-scale datasets”, Comput. Vis. Image Understand. (2013), http://dx.doi.org/10.1016/j.cviu.2013.09.004, Elsevier.

[4] Xiaozheng Zhang, Yongsheng Gao, “Face recognition across pose: A review”, Pattern Recognition 42 (2009) 2876 – 2896, Elsevier.doi: 10.1016/j.patcog.2009.04.017

[5] Nathan Intrator, Daniel Reisfeld, Yehezkel Yeshurun, “Face recognition using a hybrid supervised/unsupervised neural network”, Pattern Recognition Letters 17 (1996) 67-76, Elsevier.

[6] E. Breck, Y. Choi, and C. Cardie. Identifying Expressions of Opinion in Context. In Proceedings of Twentieth International Joint Conference on Artificial Intelligence (IJCAI), 2007.

[7] D. Fensel, F. van Harmelen, I. Horrocks, D. L. Mcguinness, and P. F. Patel-Schneider. Oil: An Ontology Infrastructure for The Semantic Web. Intelligent Systems, IEEE [see also IEEE Intelligent Systems and Their Applications], 16(2):38–45,2001.

[8] Anitra Babic, Huijun Xiong, Danfeng Yao and Liviu Iftode, “Building Robust Authentication Systems With Activity-Based Personal Questions” SafeConfig’09, November 9, 2009, Chicago, Illinois, USA.

[9] Janani Sriram, Minho Shin, Tanzeem Choudhury, David Kotz, ICMI-MLMI’09, November 2–4, 2009, Cambridge, MA, USA. “Activity-aware ECG-based patient authentication for remote health monitoring”

[10] Sheela Shankar, Dr. V. R Udupi, “International Journal of Advance Research in Computer Science and Management Studies”, A Review on the Challenges Encountered in Biometric Based Authentication.

Appendix A

Partial Java Program for Generating Public Key / Private Key Pair

```
import java.math.*;
 import java.util.*;
 import java.sql.*;

 public class GenerateKey{

 public static void main(String[] args) throws Exception {

       BigInteger p = BigInteger.probablePrime(5, new
           Random());
       BigInteger q = BigInteger.probablePrime(5, new
           Random());

       BigInteger one=new BigInteger("1");

       BigInteger n=p.multiply(q);

       BigInteger p1=p.subtract(one);
       BigInteger q1=q.subtract(one);
       BigInteger pn=p1.multiply(q1);

       System.out.println("n : " + n);
       System.out.println("pn : " + pn);

       //Find prime numbers less than pn

       int ipn=pn.intValue();
       int i;
       boolean flag;
       int arr[]=new int[500];
       int count=0;

       for (i=3;i<ipn;i++)
       {
         flag=true;
    for(int j=2;j<i;j++)
    {
      if ((i % j)==0 || (i % ipn)==0)
      {
       flag=false;
       continue;
      }
    }
    if (flag)
    {
     arr[count++]=i;
    }
  }

  boolean found;
   found=false;
   int d=0;
   long e=0;
for (int k=0;k<count;k++)
{
       e=arr[k];
   for(i=1;i<ipn;i++)
```

 ISSN: 2321-8169
6175 - 6184

```
  {
   if ((i*e)%ipn==1)
   {
      found=true;
      break;
   }
  }
  if (found)
  {
   d=i;
   break;
  }

}

  if (!found)
   System.out.println("Unable to compute private key");

  System.out.println("Public Key [" + e + "," + n + "]");
  System.out.println("Private  Key  [" + d + "," + n + "]");

}
      }
```