

# MUTATION AND CROSSOVER ISSUES FOR OSN PRIVACY

C. Narasimham<sup>1</sup>, Jacob<sup>2</sup>

<sup>1</sup> II Year M.Tech-CSE, <sup>2</sup> Assoc Prof. of CSE

Nova College of Engineering, Jangareddy Gudem, AP, India

---

**Abstract:** Privacy is one of the friction points that emerge when communications get mediated in Online Social Networks (OSNs). Different communities of computer science researchers have framed the ‘OSN privacy problem’ as one of surveillance, institutional or social privacy. In this article, first we provide an introduction to the surveillance and social privacy perspectives emphasizing the narratives that inform them, as well as their assumptions and goals. This paper mainly addresses visitors events(population) on an users account and updates the account holders log information. And thus the evolutionary aspects of Surveillance are reflected in User's Log, this needs the implementation of Genetic Algorithm. Further, this requires a bridge module between every interaction between the user and social network server. This paper implements mutation aspects through Genetic Algorithm by differing users into Guests and Friends, and identifies and Cross over issues of a guest Clicking Friend of a friend.

---

## I. INTRODUCTION

The existing work could model and analyze access control requirements with respect to collaborative authorization management of shared data in OSNs. The need of joint management for data sharing, especially photo sharing, in OSNs has been recognized by the recent work provided a solution for collective privacy management in OSNs. Their work considered access control policies of a content that is co-owned by multiple users in an OSN, such that each co-owner may separately specify her/his own privacy preference for the shared content. After careful analysis the system has been identified to have the following stages:

- ❖ The Social Privacy
- ❖ The Surveillance
- ❖ The Institutional Privacy
- ❖ Privacy As Protection

### **Social Privacy:**

Social privacy relates to the concerns that users raise and to the harms that they experience when technologically mediated communications disrupt social boundaries. The users are thus “consumers” of these services. They spend time in these (semi-)public spaces in order to socialize with family and friends, get access to information and discussions, and to expand matters of the heart as well as those of belonging. That these activities are made public to ‘friends’ or a greater audience is seen as a crucial component of OSNs. In Access Control, solutions that employ methods from user modeling aim to develop “meaningful” privacy settings that are intuitive to use, and that cater to users’ information management needs.

### **Surveillance Module:**

With respect to surveillance, the design of PETs starts from the premise that potentially adversarial entities operate or monitor OSNs. These have an interest in getting hold of as much user information as possible, including user-generated

content (e.g., posts, pictures, private messages) as well as interaction and behavioral data (e.g., list of friends, pages browsed, ‘likes’). Once an adversarial entity has acquired user information, it may use it in unforeseen ways – and possibly to the disadvantage of the individuals associated with the data.

### Institutional Privacy:

The way in which personal control and institutional transparency requirements, as defined through legislation, are implemented has an impact on both surveillance and social privacy problems, and vice versa. Institutional privacy studies ways of improving organizational data management practices for compliance, e.g., by developing mechanisms for information flow control and accountability in the back end. The challenges identified in this paper with integrating surveillance and social privacy are also likely to occur in relation to institutional privacy, given fundamental differences in assumptions and research methods.

### Privacy As Protection:

The goal of PETs (“Privacy Enhancing Technologies”) in the context of OSNs is to enable individuals to engage with others, share, access and publish information online, free from surveillance and interference. Ideally, only information that a user explicitly shares is available to his/her intended recipients, while the disclosure of any other information to any other parties is prevented. Furthermore, PETs aim to enhance the ability of a user to publish and access information on OSNs by providing her with means to circumvent censorship.

### Proposed System:

We distinguish three types of privacy problems that researchers in computer science have to tackle. The first approach addresses the “surveillance problem” that arises when the personal information and social interactions of OSN users are leveraged by governments and service providers. The second approach addresses those problems that emerge through the necessary renegotiation of boundaries as social interactions get mediated by OSN services, in short called “social privacy”. The third approach addresses problems related to users losing control and oversight over the collection and processing of their information in OSNs, also known as “institutional privacy”. This paper implements mutation aspects through Genetic Algorithm by differing users into Guests and Friends, and identifies and Cross Over issues of a guest Clicking Friend of a friend.

## II. DESIGN

The system that we propose to describe in using Object Oriented Modeling Diagrams.

### Use-case Diagram:

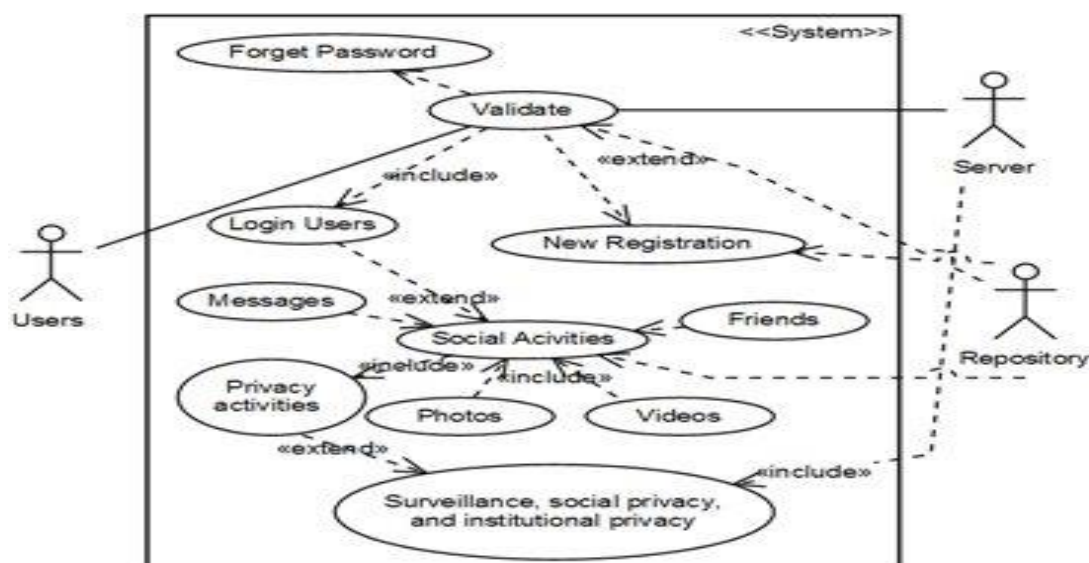


Fig 1: Use Case Diagram

**Class Diagram:**

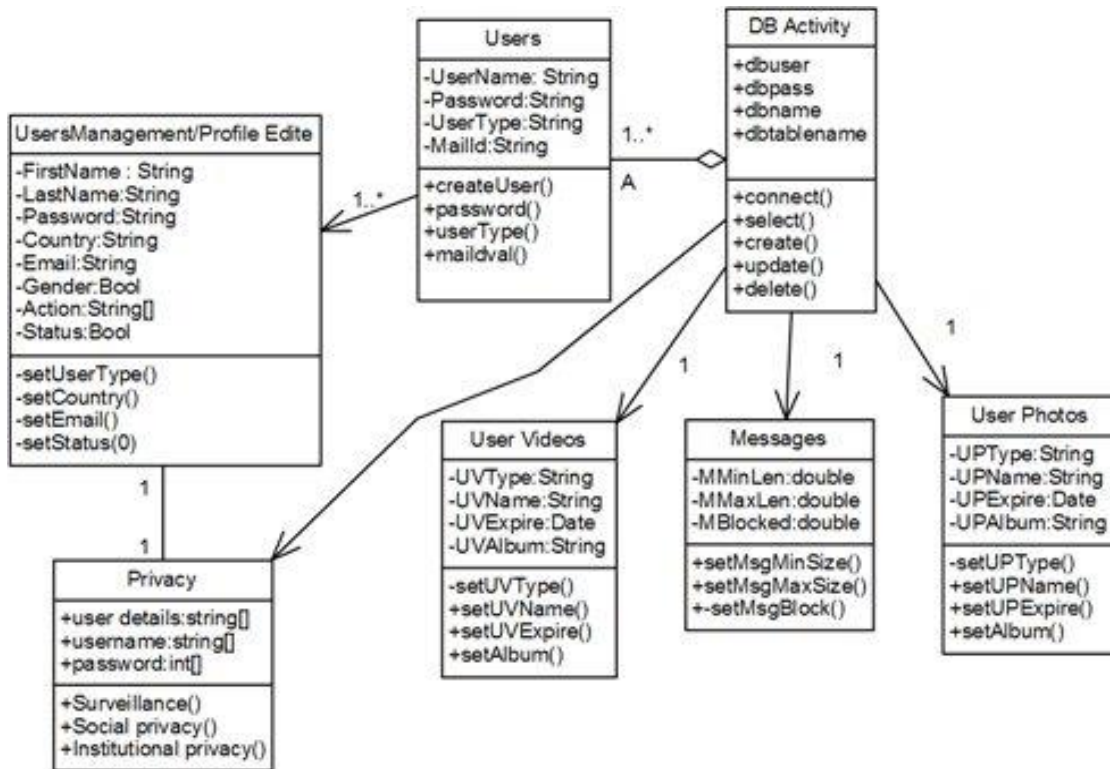


Fig 2: Class Diagram

**Sequence Diagram:**

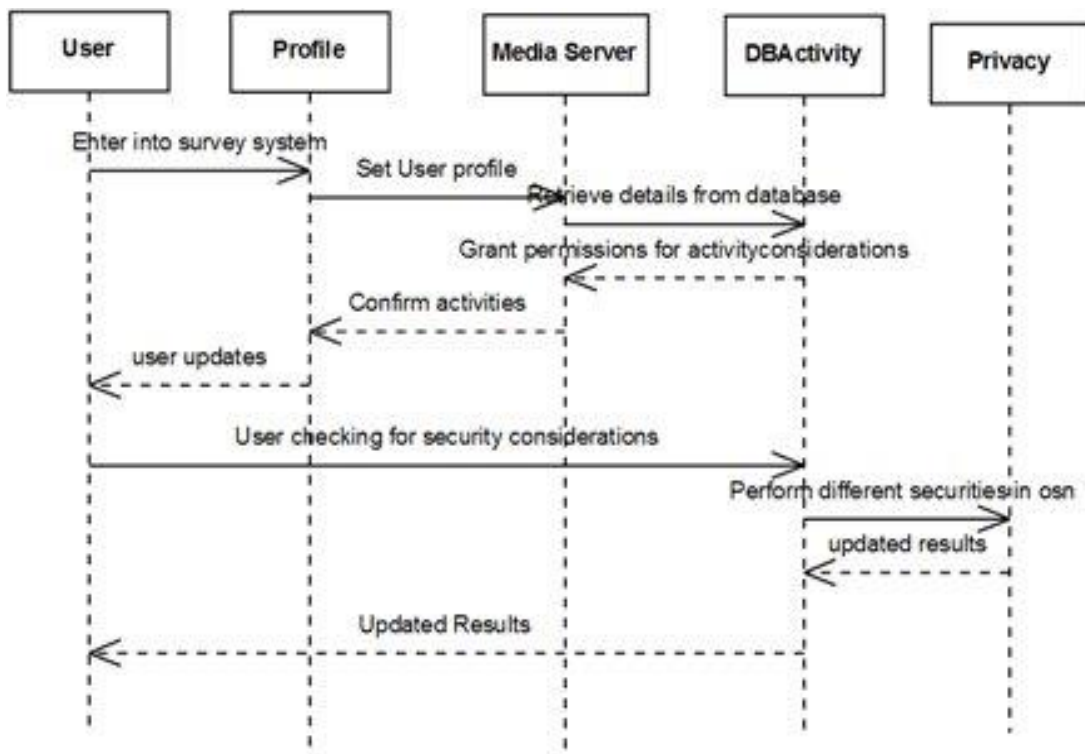
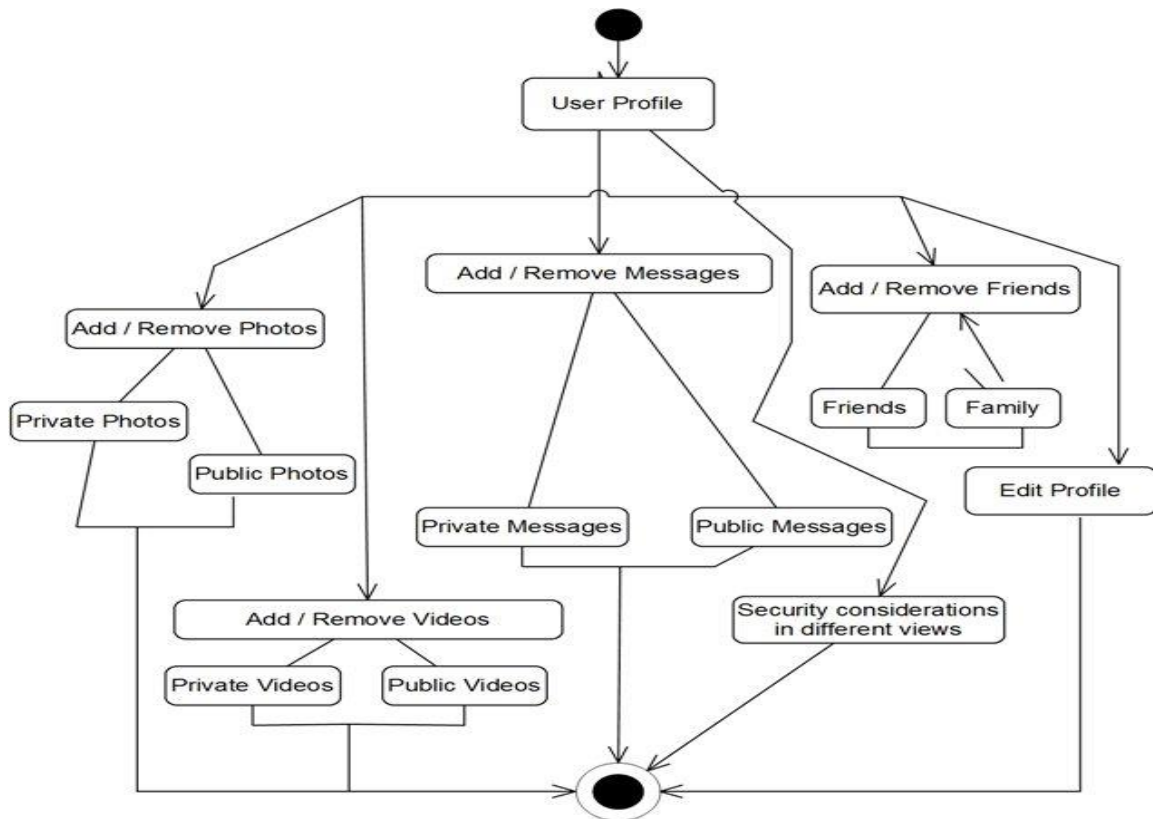


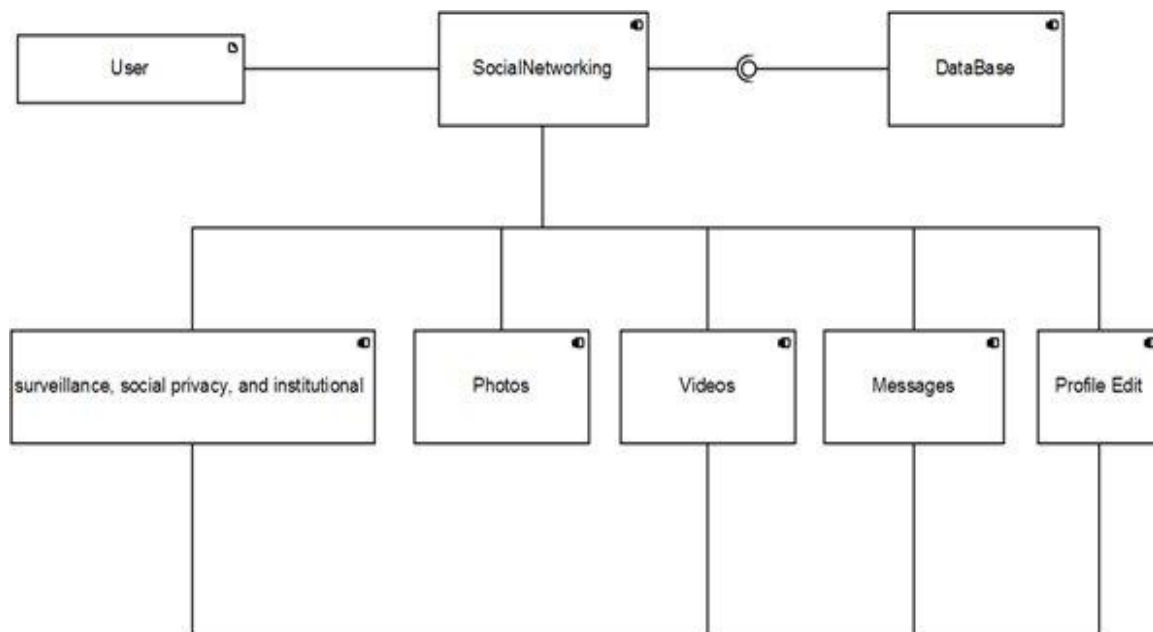
Fig 3: Sequence Diagram

**Activity Diagram:**



**Fig 4: Activity Chart**

**Component Diagram:**



**Fig 5: Component Diagram**

### Deployment Diagram:

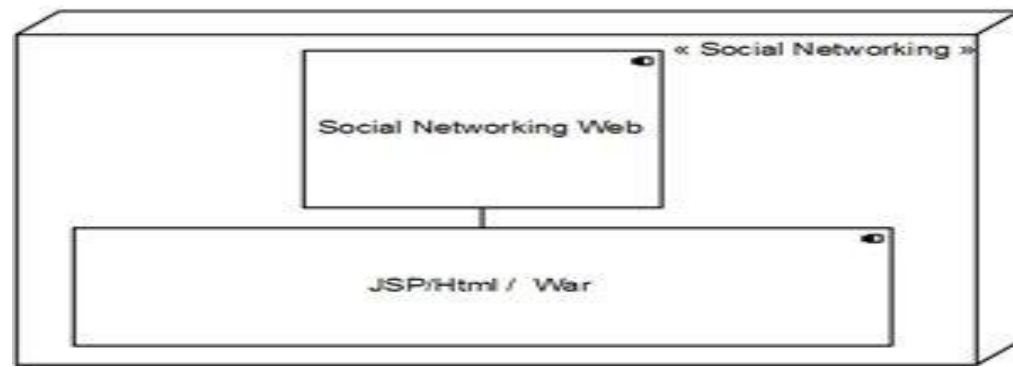


Fig: 6. Deployment Diagram

### III. IMPLEMENTATION

This module collects visitors events (population) on an user's account and updates the account holder's log information. And thus the evolutionary aspects of Surveillance are reflected in User's Log. Thus completing GA's implementation public static void saveActions (String actioned, String actioner, String action, String actionertype, String sensitiveaction) throws ClassNotFoundException, SQLException {

```
    if(actioned.trim().equals(actioner.trim()))
```

```
    { return; }
```

```
    DataBase dbfunc = new DataBase();
```

```
    dbfunc.createConnection();
```

```
String query = "insert into actionsmonitor(`actioned`,`actioner`,`action`,`actionon`,`actionertype`,`sensitiveaction`)
values(" + actioned.trim() + "," + actioner + "," + action + ",now()," + actionertype + "," + sensitiveaction + ")";
```

```
System.out.println(query);
```

```
dbfunc.updateRecord(query);
```

```
dbfunc.closeConnection();
```

```
}
```

This happens to be a bridge module between every interaction between the user and social network server. This module implements mutational aspects of GA by differing users into Guests and Friends. This module implements and identifies Cross Over Activities of a guest Clicking Friend of a friend of a friend.

```
public class Surveillance extends HttpServlet {
```

```
    * Processes requests for both HTTP
```

```
    * <code>GET</code> and
```

```
    * <code>POST</code> methods.
```

```
    * @param request servlet request
```

```
    * @param response servlet response
```

```

* @throws ServletException if a servlet-specific error occurs
* @throws IOException if an I/O error occurs
*/
protected void processRequest(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();
    try {
        HttpSession session = request.getSession(false);
        String store = (String) session.getAttribute("id");
        String email = (String) session.getAttribute("email");
        String name = (String) session.getAttribute("name");
        String option = request.getParameter("option");
        if (option.trim().equals("2")) {
            String friendMail = request.getParameter("friendMail");
            Globals.saveActions(friendMail, email, "ViewedProfile", "GUEST", "YES");
            String page = "viewProfile.jsp?option=2&friendMail=" + friendMail;
            response.sendRedirect(page);
        }
        if (option.trim().equals("3")) {
            String friendMail = request.getParameter("friendMail");
            Globals.saveActions(friendMail, email, "ViewedProfile", "FRIEND", "NO");
            String page = "viewProfile.jsp?option=1&friendMail=" + friendMail;
            response.sendRedirect(page);
        }
        // 4 users's page
        if (option.trim().equals("4")) {
            String friendMail = request.getParameter("friendMail");
            Globals.saveActions(friendMail, email, "AccountAccess", "FRIEND", "NO");
            String page = "template.jsp?option=1&friendMail=" + friendMail;
            response.sendRedirect(page);
        }
        // 5 users's wall
        if (option.trim().equals("5")) {
            String friendMail = request.getParameter("friendMail");

```

```

    Globals.saveActions(friendMail, email, "WallAccess", "FRIEND", "NO");
    String page = "template.jsp?option=2&friendMail=" + friendMail;
    response.sendRedirect(page);
}
// 6 users's photos
if (option.trim().equals("6")) {
    String friendMail = request.getParameter("friendMail");
    Globals.saveActions(friendMail, email, "ImageAccess", "FRIEND", "NO");
    String page = "template.jsp?option=3&friendMail=" + friendMail;
    response.sendRedirect(page);
}
// 7 users's videos
if (option.trim().equals("7")) {
    String friendMail = request.getParameter("friendMail");
    Globals.saveActions(friendMail, email, "VideoAccess", "FRIEND", "NO");
    String page = "template.jsp?option=4&friendMail=" + friendMail;
    response.sendRedirect(page);
}
// 8 users's friends
if (option.trim().equals("8")) {
    String friendMail = request.getParameter("friendMail");
    Globals.saveActions(friendMail, email, "FriendAccess", "FRIEND", "NO");
    String page = "template.jsp?option=5&friendMail=" + friendMail;
    response.sendRedirect(page);
}
// 9 chain users's page
if (option.trim().equals("9")) {
    String source = request.getParameter("source");
    String friendMail = request.getParameter("friendMail");
    Globals.saveActions(friendMail, email, "FriendAccess:Source("+source+") and Friend("+friendMail+")",
    "FRIEND", "NO");
    String page = "template.jsp?option=1&friendMail=" + friendMail;
    response.sendRedirect(page);
}
} catch (ClassNotFoundException ex) {

```

```

    Logger.getLogger(Surveillance.class.getName()).log(Level.SEVERE, null, ex);
  } catch (SQLException ex) {
    Logger.getLogger(Surveillance.class.getName()).log(Level.SEVERE, null, ex);
  } finally {
    out.close();
  }
}

// <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
/**
 * Handles the HTTP
 * <code>GET</code> method.
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
  processRequest(request, response);
}

* Handles the HTTP
* <code>POST</code> method.
*
* @param request servlet request
* @param response servlet response
* @throws ServletException if a servlet-specific error occurs
* @throws IOException if an I/O error occurs
*/
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
  processRequest(request, response);
}

* Returns a short description of the servlet.
* @return a String containing servlet description

```



```
@Override
public String getServletInfo() {
return "Short description";
} // </editor-fold>
}
```

#### IV. RESULTS

##### Profile Created Successfully

osn@gmail.com

Email: narasimham\_c@yahoo.com

Event Name : Profile Creation

Date : 28-10-2014 16:22:54

IP Address : 0:0:0:0:0:0:1

...Thank You...Please Visit Us For Social Networking..

##### New Image Upload

User Id : gvravindranath@gmail.com

Event Name : User (gvravindranath@gmail.com) Uploaded New Image:Jellyfish.jpg

Activity : New Image Upload

Activity On : 2014-10-28 16:19:47.0

Sent On : 28-10-2014 16:20:44

...Thank You...Please Visit Us...

##### Profile Picture Up dation

User Id : gvravindranath@gmail.com

Event Name : User(gvravindranath@gmail.com) Updated Their Profile Pic

Activity : Profile Picture Updation

Activity On : 2014-10-28 16:18:40.0

Sent On : 28-10-2014 16:19:35

...Thank You...Please Visit Us...

##### New Profile Update

osn@gmail.com

User Id : gvravindranath@gmail.com

Event Name : User (gvravindranath@gmail.com) Uploaded New Image: Koala.jpg

Activity : New Image Upload

Activity On : 2014-10-28 16:19:47.0

Sent On : 28-10-2014 16:20:35

...Thank You...Please Visit Us...

User Id : gvravindranath@gmail.com

Event Name : User (gvravindranath@gmail.com) updated Their Profile

Activity : New Profile Update

Activity On : 2014-10-28 16:17:20.0

Sent On : 28-10-2014 16:17:35

...Thank You...Please Visit Us...

## V. CONCLUSION

Privacy is one of the friction points in Online Social Networks (OSNs). In this article, we conclude that the surveillance and social privacy perspectives have been emphasized. And, addressed the visitors events on an users account and updated with the account holders log information. This paper described the implementation of a bridge process between every interaction between the user and social network server. Further, implemented mutational aspects of Genetic Algorithm by differing users into Guests and Friends, and identified Cross Over Activities of a guest Clicking Friend of a friend of a friend.

## REFERENCES

- [1] Seda Gurses, Claudia Diaz “Two Tales of Privacy in Online Social Networks” IEEE Security and Privacy Vol: PP No 99, 2013.
- [2] J. Anderson, C. Diaz, J. Bonneau, and F. Stajano. Privacy-Enabling Social Networking over Untrusted Networks. In ACM Workshop on Online Social Networks (WOSN), pages 1–6. ACM, 2009.
- [3] F. Beato, M. Kohlweiss, and K. Wouters. Scramble! your social network data. In Privacy Enhancing Technologies Symposium, PETS 2011, volume 6794 of LNCS, pages 211–225. Springer, 2011.
- [4] E. De Cristofaro, C. Soriente, G. Tsudik, and A. Williams. Humming-bird: Privacy at the time of twitter, IEEE Symposium on Security and Privacy, pages 285–299. IEEE Computer Society, 2012.
- [5] .A. Cutillo, R. Molva, and T. Strufe. Safebook: A privacy-preserving online social network leveraging on real-life trust. Communications Magazine, 47(12):94–101, 2009.
- [6] James Grimmelmann. Saving Facebook. Iowa Law Review, 94:1137– 1206, 2009.
- [7] Kevin D. Haggerty and Richard V. Ericson. The Surveillant Assemblage. British Journal of Sociology, 51(4):605 – 622, 2000.
- [8] Evgeny Morozov. Face book and Twitter are just places revolutionaries go. The Guardian, 11. March 2011.
- [9] Deirdre K. Mulligan and Jennifer King. Bridging the gap between privacy and design. Journal of Constitutional Law, 14 (4):989 – 1034, 2012.