Implementation of the Role Based Access Control in Application for Mobile Device on the Android OS Platform*

Diasamidze S. V., Kuzmenkova E. Yu., Kuznetsov D. A., Sarkisyan A. R.
Petersburg State Transport University
St. Petersburg, Russia
sv.diass99@ya.ru

Abstract. **The article is dedicated to solving the problem of privacy of personal data stored on the mobile device. We consider the method of the personal information protection based on role-based access control, it's advantages and software implementation on the Android OS platform.**

Keywords: **Information security, Data security, Role based access control, Confidentiality.**

Introduction

Modern development of information technologies and the spread of mobile devices has led to the fact that modern mobile device – smartphone or tablet – is used as a universal personal device, that includes mobile office, entertainment center and Internet work tool. There is a huge amount of personal information stored in memory of the smartphone: contacts of your colleagues, friends and relatives with their personal data; call history; corporate correspondence; settings of the Wi-Fi access points, which are located within the habitat of the owner; social network applications (often with saved passwords); Bank details or mobile/SMS banking, photos, videos, notes, etc. This concentration of business and personal data leads to the fact that the abstract value of information outweighs the price of the device itself. That is why the task of information protection mobile device is extremely important [1, 2].

This article describes one way of solving the problem of confidentiality of personal data stored on personal mobile device.

The problems of security

One of the security problems appear in situation when it is necessary to provide the stranger with a temporary access to the device. For example, after purchasing a new smartphone acquaintances or friends ask to see and appreciate it. In this situation, on the one hand, it is uncomfortable to refuse them, and it would be wrong to give outsiders access to confidential information stored on your phone, such as private messages, photos and videos, accounts, contacts, etc. The solution to this problem would be to create a guest account, which would have limited access to the above resources.

Also it often happens that a child (son, daughter, younger brother or sister) asks to play games on your phone or tablet. However there is a possibility that while operating the phone he will exit the game, and view different application, accidentally reset settings or delete any important information. You can provide a limited account, which will have available only game apps.

Currently on the market there are apps with similar functionality, such as AppLock, Leo Privacy Guard, etc. But they contain other principles of limiting access to data. You have the option to set separate passwords for each application that from our point of view is not convenient and appropriate. Also, as a result of our testing it became clear that if you turn off your phone, the settings of these applications are discarded, that violates the data protection principles. In our application, Allock we tried to eliminate such errors.

Certainly, as with any role-based access control the main will be the account of the device owner, who is granted with full access rights to all resources and applications.

Role based access control

As a base mechanism we selected the role-based access control. It's basic idea is to maximize the approximation of the system logic to the actual separation of staff's functions in the organization, which means that the method of role-based access control monitors user access to information based on the types of their activities in the system. The subjects' rights of access to the objects are grouped considering the specifics of their application, forming a role.

To analyze and study properties of systems role-based access right the mathematical models are used [3].

The base model of role-based access control defines the main principles and elements of such models.

Main elements:

U – set of users;

R – set of roles;

P – set of access rights on the objects of the computer system;

S – set of user's sessions;

PA – function, that defines a set of access rights for each role: PA: $R \to 2^P$, wherein for each $p \in P$ exists $r \in R$ such that $p \in PA(r)$.

UA – function, that defines for each user a set of roles that he can be authorized: UA: $U \to 2^R$.

For completeness of the mathematical model the basic functionsare introduced:

user: $S \to U$ – function that determines for each user session on behalf of which it's activated.

roles: $S \to 2^R$ – function that defines for the user a set of roles to which he can be authorized in this session.

In the basic model of role-based access control the following rules are defined:

* This article is published with the support of Federal state budget educational institution of higher professional education "Petersburg State Transport University of Emperor Alexander I" of initiative research works of student research teams.

1. One subject can have multiple roles.
2. One role can have multiple subjects.
3. One role can have multiple permissions.
4. One permission can belong to multiple roles [4].

One of the important mechanisms of the basic model of access rights differentiation are restrictions on many roles that can be authorized by the user, or to which he authorizes in the same session. This mechanism is also necessary for widespread use of the basic model of role differentiation, because it provides better conformity toused in computer systems technologies of data processing [5].

The relationships between the structural elements of the basic model role-based access rights, as shown in fig. 1.

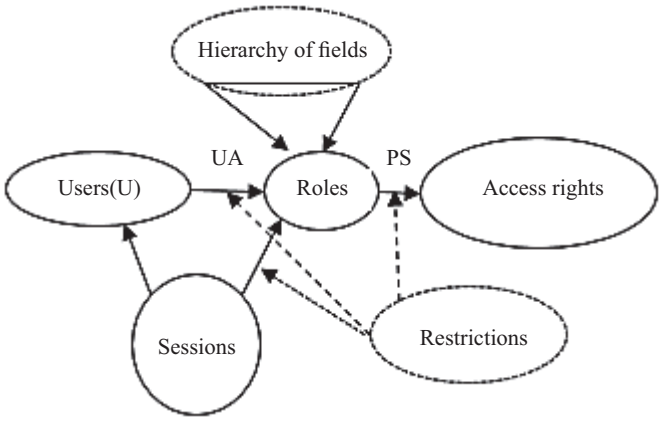

Fig. 1. The structure of the role-based access control

The simplified model of role-based access control can be represented as follows (fig. 2).

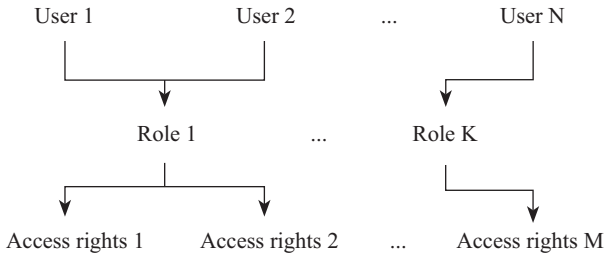

Fig. 2. The simplified model of role-based access control

The advantages of the role-based access control

Role-based access control is neutral to specific types of rights and methods of their verification; it can be considered as an object oriented framework that facilitates administration, because it enables you to make subsystem access controlled by arbitrarily large number of users, primarily through the establishment of relationships between roles, similar to inheritance in object oriented systems. In addition, the number of roles should be much less than users. As a result, the number of administered connections becomes proportional to the sum (not the multiplication) of the number of users and objects that is impossible to decreasein the order of magnitude [6].

While developing applications for mobile devices model of the role-based access rights is the most convenient, reliable to protect user's personal data. It has a number of advantages over the other security policies.

First, the simplicity of administration. In the classical models of access control the rights to perform certain operations on the object are registered for each user or group of users. In role model the separation of the concepts "role" and "user" allows you to break the task into two parts: the definition of user roles and defining the rights of access to the object for the role. This approach greatly simplifies the process of administration, because when you change the scope of responsibility of the user it is enough to remove his old role and assign the other corresponding to his new rights. For example, in Allock for the role "Guest" at first was installed access to five gaming applications, but the owner at any time may extend or narrow the range of access of this user [7].

Secondly, the principle of least privilege. A role model allows the user to register in the system with minimal role that allows him to perform the required tasks. Users with multiple roles, do not always require all possible privileges to perform specific tasks. According to the principle of least privilege, the user receives only those access rights which he needs to perform a specific task. This requires to clarify the objectives of the task, the set of privileges required to execute and restrict user privileges by this set. Prohibition of user privileges that are not required to perform the task, avoids opportunities to circumvent the system security policy. That is, the user in role "Guest" can not access your correspondence or your schedule for the day/week, phone settings, he has only the access to the function "make a call", "send SMS", as well as access to some gaming applications.

The implementation of the application

While adapting mechanism of differentiation of access rights based on roles we were asked to identify some typical roles with pre-defined powers, such as "Owner", "Guest", "Children". The user is also given the opportunity to introduce new roles and define their access rights to resources of the mobile device

As an environment for developing application was used Android Studio with minimum support operating system version 4.0, which based on the published statistics, allows you to use the app on 99 % of active devices under Android OS (fig. 3) [8].

Version	Codename	API	Distribution
2.2	Froyo	8	0.3%
2.3.3 - 2.3.7	Gingerbread	10	5.6%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	5.1%
4.1.x	Jelly Bean	16	14.7%
4.2.x		17	17.5%
4.3		18	5.2%
4.4	KitKat	19	39.2%
5.0	Lollipop	21	11.6%
5.1		22	0.8%

Fig. 3.Statistics of Android version using. March 2015

The design meets the guidelines of (the manufacturer's instructions on formalization) Google material design, supported from version 5.0.

General algorithm of the application looks as follows:

1. Installation of the application Allock on a mobile device, then the main menu is automatically loaded (fig. 4).

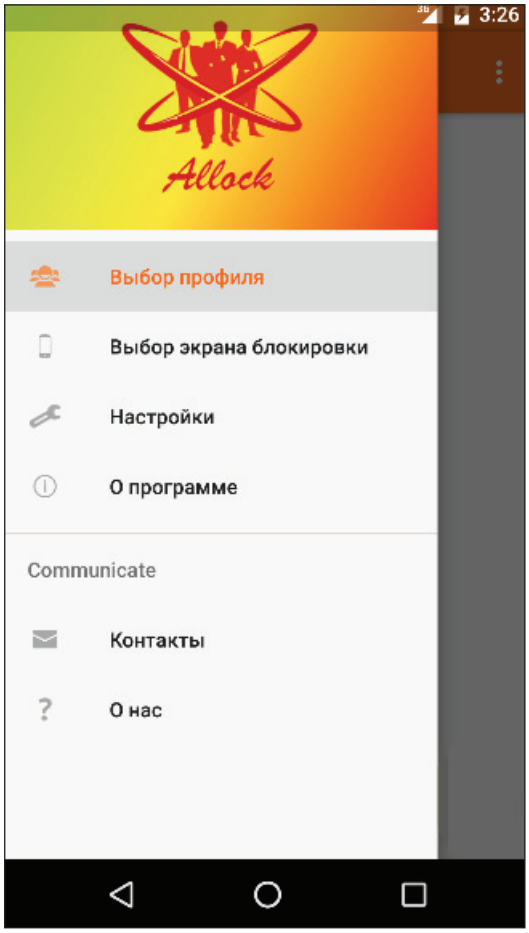

Fig. 4. Main menu of the Allock application

2. Selecting and customizing access for one of the model groups, by defaultgroups such as «Owner», «Guest», «Children» are installed in application with appropriate access rights (fig. 5). You can create a new group at the discretion of the owner of the device.

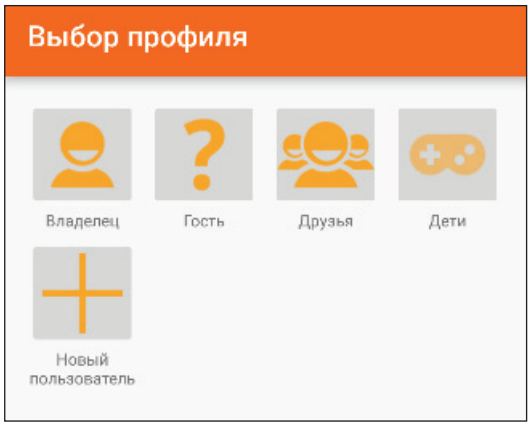

Fig. 5. Menu of group choice

3. Working with the list of installed applications. After selecting a specific group there will appear a list of all installed applications. The applicationsthat would be available for the user authorized under account of this groupshould be selected in the appropriate field [6].

4. Setting a unique password for the group. To increase the level of security in the Allock application there is a uniqueness rule of used authenticators -passwords for each group should be different. The measure was introduced in the absence of the name of the group on the lock screen for more simple use the application (the user needs only to enter a password, which is an identifier and authenticator of the group at the same time).

For receiving a list of all installed applications we use classes *UserApps.java* (fig. 6) and *AppAdapter.java* (fig. 7). *UserApps.java*is responsible for the build of screen form (fig. 8), and as we get the list of applications in this class, then this class is inherited from *ListActivity* class. The important methods are *checkForLaunchIntent* and class *LoadApplications* (fig. 9), which loadtheappsinstalled on device. *AppAdapter.java* is the class adapter for *UserApps.java.* The adapter isused to build some dynamic data (in our case it is the listwith unknown length) of objects with information about applications. It takes the provided data and places them in order, immediately setting the described components (we have this *ImageView*, a *TextView* for the icons and app name).

The key method is *getView* in which the process described abovetakes place.

Conclusion

The important point in the development of the app was to create our own lock screen, attached to the all the installed profiles using the password as identifier-authenticator. The screen extends the capabilities of standard one, allows you to recognize a few passwords and activate the appropriate account (group) with a specific set of authorized applications.

The advantages of our developed program are simplicity of use, automation of complicated processes of access rightsdifferentiation that is absolutely imperceptible for users, flexibility and easiness of settings.

References

1. Artamonov V.A. Security problems of mobile devies, systems and applications [Problemy bezopasnosti mobilnykh ustroystv, system i prilozheniy], *IT Bezopasnost [IT Security]*. Available at: http://itzashita.ru/mobilnyie-ustroystva/bezopasnost-mobilnyih-ustroystv-sistem-i-prilozheniy.

2. Yakushin P. Secutity of the mobile enterprise [Bezopasnost mobilnogo predpriyatiya], *Otkryt yesistemy. SUBD [Open systems. SUBD]*, 2013, no. 1, pp. 22-26.

3. Devyanin P.N. *Modeli bezopasnosti computernykh sistem. Upravlenie dostupom i informatsyonnymi potokami* [Security models of computer systems. Control of access and informational threads], 2nd ed., Moscow, Goryachaya liniya – Telekom, 2013, 338 p.

4. *Upravlenie dostupom na osnove roley* [Role based access control]. Available at: https://ru.wikipedia.org/wiki/Управление_доступом_на_основе_ролей (accessed 11 Feb. 2016).

```
public class UserApps extends ListActivity {

    private PackageManager packageManager = null;
    private List applist = null;
    private AppAdapter listadapter = null;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.user_apps);
        packageManager = getPackageManager();
        new LoadApplications().execute();
    }

    public void onClickStart(View view) {
        Intent intent = new Intent(this, LoginActivity.class);
        startActivity(intent);
    }

    private List<ApplicationInfo> checkForLaunchIntent(List<ApplicationInfo> list) {

        ArrayList appList = new ArrayList();
        for(ApplicationInfo info : list) {
            try{
                if(packageManager.getLaunchIntentForPackage(info.packageName) != null) {
                    appList.add(info);
                }
            } catch(Exception e) {
                e.printStackTrace();
            }
        }

        return appList;
    }
```

Fig. 6. Class UserApps

```
public class AppAdapter extends ArrayAdapter{

    private List appList = null;
    private Context context;
    private PackageManager packageManager;

    public AppAdapter(Context context, int resource,
                      List objects) {
        super(context, resource, objects);
        this.context = context;
        this.appList = objects;
        packageManager = context.getPackageManager();
    }

    public View getView(int position, View convertView, ViewGroup parent) {
        View view = convertView;

        if(null == view) {
            LayoutInflater layoutInflater = (LayoutInflater) context
                    .getSystemService(Context.LAYOUT_INFLATER_SERVICE);
            view = layoutInflater.inflate(R.layout.list_item, null);
        }
        ApplicationInfo data = (ApplicationInfo) appList.get(position);

        if(null != data) {
            TextView appName = (TextView) view.findViewById(R.id.app_name);
            ImageView iconView = (ImageView) view.findViewById(R.id.app_icon);

            appName.setText(data.loadLabel(packageManager));
            iconView.setImageDrawable(data.loadIcon(packageManager));
        }
        return view;
    }
}
```

Fig. 7. Class AppAdapter

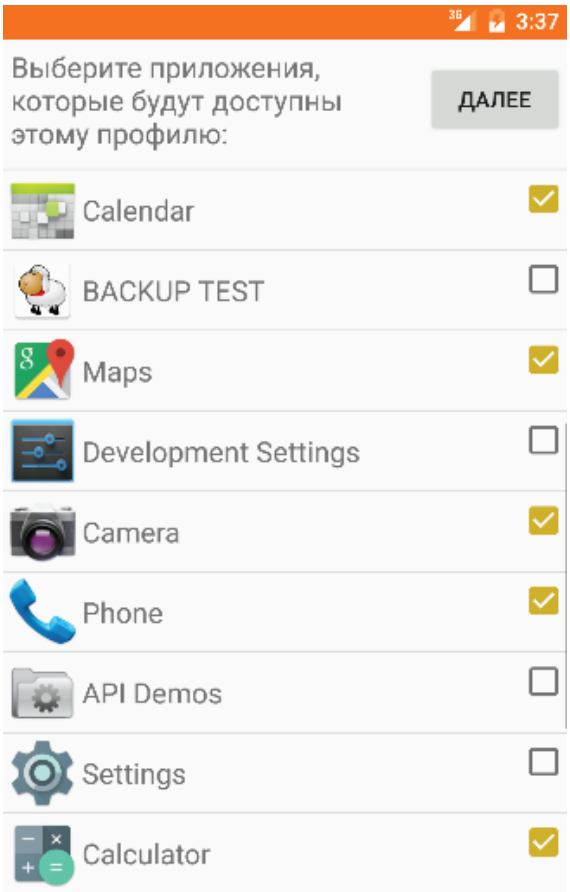

Fig. 8. Working with the list of installed applications

```
private class LoadApplications extends AsyncTask<Void, Void, Void> {

    private ProgressDialog progress = null;

    @Override
    protected Void doInBackground(Void... params) {

        applist = checkForLaunchIntent
                (packageManager.getInstalledApplications(PackageManager.GET_META_DATA));
        listadapter = new AppAdapter(UserApps.this, R.layout.list_item, applist);
        return null;
    }
}
```

Fig. 9. Class LoadApplications

5. Stepanenko I. D. *Modeli upravleniya dostupom: diskretnaya, mandatnaya, rolevaya* [Models of access control: discrete, mandatory, role-based]. Available at: http://gman1990.ru/articles.php?article_id=83 2015.

6. *Ponyatie I naznachenie modeli bezopasnosti* [Concept and purpose of security model]. Available at: http://www.studfiles.ru/preview/2880350/page:4 (accessed 11 Feb. 2016).

7. Gaydamakin N.A. *Razgranichenie dostupa k informatsii v komputernykh sistemakh* [Access distinction to information in computer systems], Ekaterinburg, Ural Univ. Publ., 2003, 328 p.

8. *Statistika versiy Andriod: mart 2015* [Android version statistics: march 2015]. Available at: http://puregoogle.ru/2015/03/03/http://puregoogle.ru/2015/03/03/statistika-versij-android-mart-2015 (accessed 20 Feb. 2016).

Реализация ролевой политики разграничения прав доступа в приложении для мобильного устройства под управлением ОС Android

Диасамидзе С. В., Кузьменкова Е. Ю., Кузнецов Д. А., Саркисян А. Р.
ФГБОУ ВО ПГУПС
Санкт-Петербург, Россия
sv.diass99@ya.ru

Аннотация. **Статья посвящена решению проблемы обеспечения конфиденциальности личных данных, хранящихся на мобильном устройстве. Рассматривается вариант метода защиты персональной информации, основанный на ролевой политике разграничения прав доступа, его преимущества и программная реализация на платформе ОС Android.**

Ключевые слова: **защита информации, безопасность данных, ролевая политика разграничения прав доступа, конфиденциальность.**

Литература

1. Артамонов В. А. Проблемы безопасности мобильных устройств, систем и приложений / В. А. Артамонов // ИТ Безопасность. – URL : http://itzashita.ru/mobilnyie-ustroystva/bezopasnost-mobilnyih-ustroystv-sistem-i-prilozheniy.

2. Якушин П. Безопасность мобильного предприятия / П. Якушин // Открытые системы. СУБД. – 2013. – № 1. – С. 22–26.

3. Девянин П. Н. Модели безопасности компьютерных систем. Управление доступом и информационными потоками : учеб. для вузов / П. Н. Девянин. – 2-е изд., перераб. и доп. – М. : Горячая линия – Телеком, 2013. – 338 с.

4. Управление доступом на основе ролей. – URL : https://ru.wikipedia.org/wiki/Управление_доступом_на_основе_ролей (дата обращения 11.02.2016).

5. Степаненко И. Д. Модели управления доступом : дискретная, мандатная, ролевая. – URL : http://gman1990.ru/articles.php?article_id=83.

6. Понятие и назначение модели безопасности. – URL : http://www.studfiles.ru/preview/2880350/page:4 (дата обращения 11.02.2016).

7. Гайдамакин Н. А. Разграничение доступа к информации в компьютерных системах / Н. А. Гайдамакин. – Екатеринбург : Изд-во Урал. ун-та, 2003. – 328 с.

8. Статистика версий Android : март 2015. – URL : http://puregoogle.ru/2015/03/03/statistika-versij-android-mart-2015 (дата обращения 20.02.2016).