

# Design of Image Verification Code Based on Android SDK

Jiajun Du, Hong Lan

School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou Jiangxi  
Email: 40376334@qq.com

Received: Mar. 9<sup>th</sup>, 2020; accepted: Mar. 20<sup>th</sup>, 2020; published: Mar. 27<sup>th</sup>, 2020

---

## Abstract

In the mobile Internet era, with the continuous development of mobile APPs, people have put forward higher requirements for the information security of APPs. The research purpose of this article is to use the Android SDK design to implement cryptographic verification and to enhance the security of information on the application to a greater extent. Using the Android SDK, this article designs an image verification code based on Android Canvas and Paint. The image verification code is generated through the steps of initialization data, target selection, angle change, weight change, color conversion, and adding interference to prevent machine recognition. The picture verification code is the most widely used verification code at present, compared with other verification codes, it is easier to operate and human eye recognition. In order to identify the picture verification code before requesting the user's account password to the server, the application security is greatly increased. The verification code not only effectively prevents brute force cracking by illegal users, but also reduces the server request pressure caused by brute force attacks.

## Keywords

Image Verification Code, Android SDK, Prevention of Machine Identification, Brute Force Cracking

---

# 基于安卓SDK的图片验证码设计

杜佳俊, 兰红

江西理工大学信息工程学院, 江西 赣州  
Email: 40376334@qq.com

收稿日期: 2020年3月9日; 录用日期: 2020年3月20日; 发布日期: 2020年3月27日

---

## 摘要

移动互联网时代, 随着手机APP的不断发展, 人们对APP的信息安全提出了更高要求。本文的研究目的

就是利用安卓SDK设计实现加密验证, 更大程度上加强应用上信息的安全性。利用安卓SDK, 本文设计了一种基于安卓Canvas和Paint的图片验证码, 通过初始化数据、目标选取、角度变化、权重变化、色彩变换、添加干扰等步骤生成图片验证码, 用于防范机器识别。图片验证码在当下是应用最广的验证码, 与其他验证码相比, 更容易操作和人眼识别。在将用户的账户密码请求到服务器之前, 先进行图片验证码的识别, 大大增加了应用的安全性。该验证码不仅有效的防止了非法用户的暴力破解, 还减轻了暴力攻击所导致的服务器请求压力。

## 关键词

图片验证码, 安卓SDK, 防范机器识别, 暴力破解

Copyright © 2020 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

验证码(CAPTCHA)是一种判断人与机的公共全自动程序。验证码的主要作用是为了防止一些黑客把密码数据化盗取以及保护用户信息。

文献[1]指出“验证码作为一种辅助安全手段在 Web 安全中有着特殊的地位,验证码安全和 Web 应用中的众多漏洞相比似乎微不足道,但是千里之堤毁于蚁穴,有些时候如果能绕过验证码,则可以把手动变为自动,对于 Web 安全检测有很大的帮助”。文献[2]和文献[3]对防范信息安全问题,阐明了“通过现有的安全技术对信息系统实施安全保护”观点。文献[4]介绍了几个大数据时代影响计算机网络安全因素,并说明了应如何做好防护措施。

现有的生成验证码的方式有很多,但主要的是如下几类:

第一类是文本验证码,采用随机字符显示在文本框内,但是这种验证码极易被机器识别,不安全。

第二类是图片验证码,有的是基于图像处理[5] [6] [7]的,采用的是对图片进行加噪处理,但是会增加图像识别的难度,会影响用户的体验感。还有的是选择多个与提示内容相关的图片[8],这类验证码的确会让机器人很难对图片识别进行归类,但是这有时候图像的不确定性会让人会感觉复杂。还有的是基于常识的图片验证码[9],采用问答的形式,但是有些常识也不是所有用户都能识别的。

第三类是滑块验证码,采用的是判断拖动块和目标块是否重合,并且根据用户在滑动滑块的一系列操作(如拖拽、停留)来判断是否是人操作,这类验证码最早是淘宝先推出的,但是有的应用因拖动块和目标块的重合算法不精密,导致用户体验差。

第四类是点触式验证码,采用的是从汉字库中选择 4 个汉字并随机分布,用户通过点击相应汉字的顺序进行识别。

本文提出了一种基于安卓 Canvas (画布)和 Paint (画笔)的图片验证码,它主要采用了随机验证码以及随机干扰物的方式,这不仅简化了算法量,还防止了机器自动识别,提高了验证码的安全性。

## 2. 相关知识介绍

### 2.1. Android SDK 简介

SDK: Software Development Kits, 软件开发工具包, 是软件开发工程师用于为特定的软件包、软件框架、硬件平台、操作系统等建立应用软件的开发工具的集合。Android SDK 就是 Android 软件开发人员

的工具包。该工具包中包含安卓平台上的各种 API，并在 Android 应用的设计上可以使用到。

## 2.2. Canvas 画布

Canvas 是 Android SDK 中提供的一个 API，主要用于保留绘制的内容。

Canvas 类的常用方法如下：

- Canvas(Bitmap.createBitmap (width, height, Bitmap.Config.ARGB\_8888))：构造方法，创建指定格式，大小的位图。
- Canvas.drawText(“绘制文本内容”，x, y, paint)：绘制的文本内容，正在绘制的文本原点的 X 坐标，正在绘制的文本原点的 Y 坐标，画笔对象。
- Canvas.drawColor(Color.WHITE)：将画布填充为白色。
- Canvas.drawLine(startX, startY, stopX, stopY, paint)：绘制出一条线。
- Canvas.save()：保存一下画布。
- Canvas.restore()：状态回滚：用 save 方法保存后，从栈顶取出一个状态然后根据内容进行恢复，也就是说将 Canvas 还原成最近的一个 save()的状态。

## 2.3. Paint 画笔

Paint 是 Android SDK 中提供的用于绘制内容的 API，Paint 类常用的方法如下：

- paint.setAntiAlias()：设置画笔抗锯齿。
- paint.setColor()：设置画笔颜色。
- paint.setFakeBoldText()：设置画笔文字粗体。

## 3. 图片验证码算法设计

### 3.1. 设计思想

随机产生  $n$  个字符，对字符进行旋转变色等变换处理，然后再随机生成干扰物影响机器的识别，最终形成图片显示在屏幕上。当用户访问登录界面的时候，验证码的验证会在对账户和密码的验证之前进行，提高应用程序的运行效率。验证码正确，则再进行对账号密码的验证，否则需要重新输入账号密码及验证码。

### 3.2. 算法设计及实现流程

首先通过随机产生的数值得到相应的字符串，然后通过 Canvas 和 Paint 类，将验证码生成一个简单的图片。在生成图片的过程中，加入干扰信息(如下划线以及字符大小、字符颜色)加强验证码的安全性。

图片验证码算法的具体流程如图 1 所示：

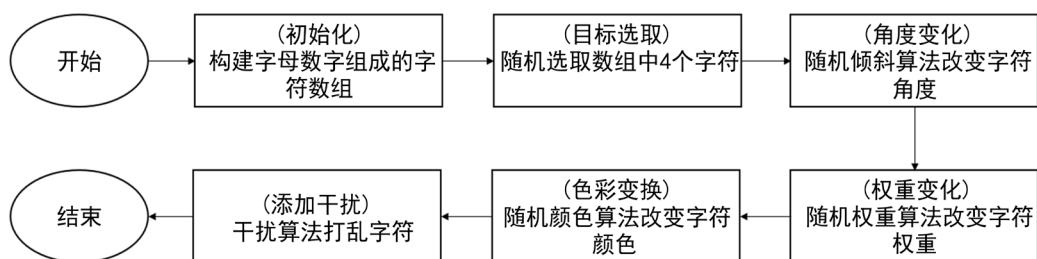


Figure 1. Algorithm flowchart

图 1. 算法流程图

### 3.3. 算法的具体实现步骤

图片验证码的实现首先获取要显示的验证码信息, 然后进行干扰设置, 干扰主要是通过随机字符样式、颜色、字符粗细与倾斜度以及随机位置设置干扰线实现, 最后再利用画布和画笔生成图片。具体实现步骤如下:

- 1) 初始化验证码数据, 定义一个一维字符数组, 由 10 个数字和 27 个字母组成。
- 2) 验证字符选取, 生成 37 以内的随机数, 随机生成字符串, 包含两步:
  - a) 通过随机函数获得  $n$  个随机数字, 对应字符数组的下标;
  - b) 获取这  $n$  个数字的数组下标对应的字符数组中的  $n$  个字符, 得到一个字符串。算法描述如下:  
字符随机选取算法伪代码描述:

---

输入: 初始化字符串 Array, 验证码长度  $n$   
输出: 随机选取的字符串(验证码)newArrayy

---

过程: //遍历  $n$  次, 即随机选取  $n$  个下标  
for  $i = 0 \rightarrow n-1$  do  
    //得到 37 以内的随机下标  
     $j \leftarrow \text{randomInt}(37)$   
    //得到随机字符  
     $\text{newArray}[j] \leftarrow \text{Array}[i]$   
end for

---

- 3) 对字符串的每个字符设置随机倾斜度, 包含两个随机变换:
  - a) 通过随机算法得到一个 double 类型的值 skew, 用于表示字符的倾斜度;
  - b) 通过随机算法得到一个布尔值, 判断这个值来决定左倾还是右倾。算法描述如下:  
字符倾斜变换算法伪代码描述:

---

输入: 验证码 newArray, 验证码长度  $n$   
输出: 倾斜变换的字符串 newArray\_1

---

过程: //遍历字符串 newArray 的每个字符  
for  $i = 0 \rightarrow n-1$  do  
    //生成随机倾斜度  
     $\text{inclination} \leftarrow \text{randomFloat}()$   
    //左倾或者右倾  
    if  $\text{randomBoolean}()$  为 true then  
        字符向左倾斜 inclination 度;  
    else  
        字符向左右斜 inclination 度;  
end for

---

- 4) 为字符串的每个字符变换字体, 设置字符的随机权重, 通过随机算法得到一个布尔值 flag, 判断这个值来决定字符是粗体还是正常。算法描述如下:

字符权重变换算法伪代码描述:

---

输入: 倾斜变化的验证码 newArray\_1, 验证码长度  $n$   
输出: 权重变化的字符串 newArray\_2

---

过程: //遍历字符串 newArray\_1 的每个字符  
for  $i = 0 \rightarrow n-1$  do  
    //粗体还是正常  
    if  $\text{randomBoolean}()$  为 true then  
        字符粗体;  
    else  
        字符正常;  
end for

---

- 5) 对字符串的每个字符设置字体颜色, 字符的随机颜色生成包含 3 步:
- 通过随机算法得到 3 个整数值, 分别代表 RGB 三色板的取值  $r$ 、 $g$ 、 $b$ ;
  - 利用 Color 类中的 `rgb(r, g, b)` 静态方法生成一个颜色值 `color`。算法描述如下:  
字符颜色变换伪代码描述:

---

```

输入: 验证码长度 n
输出: 颜色列表 colorList

过程: //遍历验证码中每个字符, 为每个字符都添加颜色变换
for i = 0 → n-1 do
    //随机生成颜色值 R、G、B
    R, G, B;
    color ← Color.rgb(R, G, B)
    colorList.add(color)
end for

```

---

- 6) 对变换后的字符串添加干扰, 随机位置设置一些干扰线。算法描述如下:  
添加干扰线伪代码描述:

---

```

输入: 干扰线数量 m
输出: 干扰线列表 lineList

过程: //遍历 m 次, 生成 m 根干扰线
for i = 0 → m-1 do
    //随机生成开始坐标 X、Y 以及结束坐标 X、Y
    stratX, startY, endX, endY;
    //绘制直线
    line ← new Line(stratX, startY, endX, endY)
    lineList.add(line)
end for

```

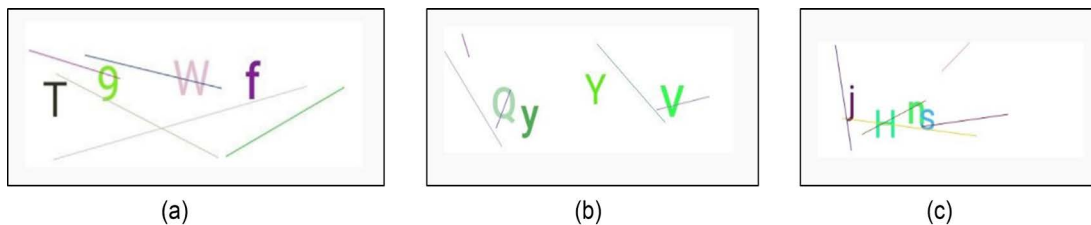
---

- (7) 在画布上生成验证码图片, 包含两步:
- 通过随机算法得到两个整形值, 用来表示文字基线的上边界 `left` 和左边界 `top`;
  - 根据字符变换的参数, 如倾斜度、字符权重等, 设置画笔;
  - Canvas 类的 `setText()` 方法在画布上画出前 6 步生成的验证码。

## 4. 运行结果及测试

### 4.1. 运行效果

在安卓 APP 中采用本文设计算法生成验证码, 效果如图 2、图 3 所示。



**Figure 2.** Canvas-based verification code generation  
**图 2.** 基于画布的本文验证码生成

通过分析算法生成的图 2(a)~(c), 可以看出字符的随机倾斜度、权重、颜色、位置、干扰线在变化。

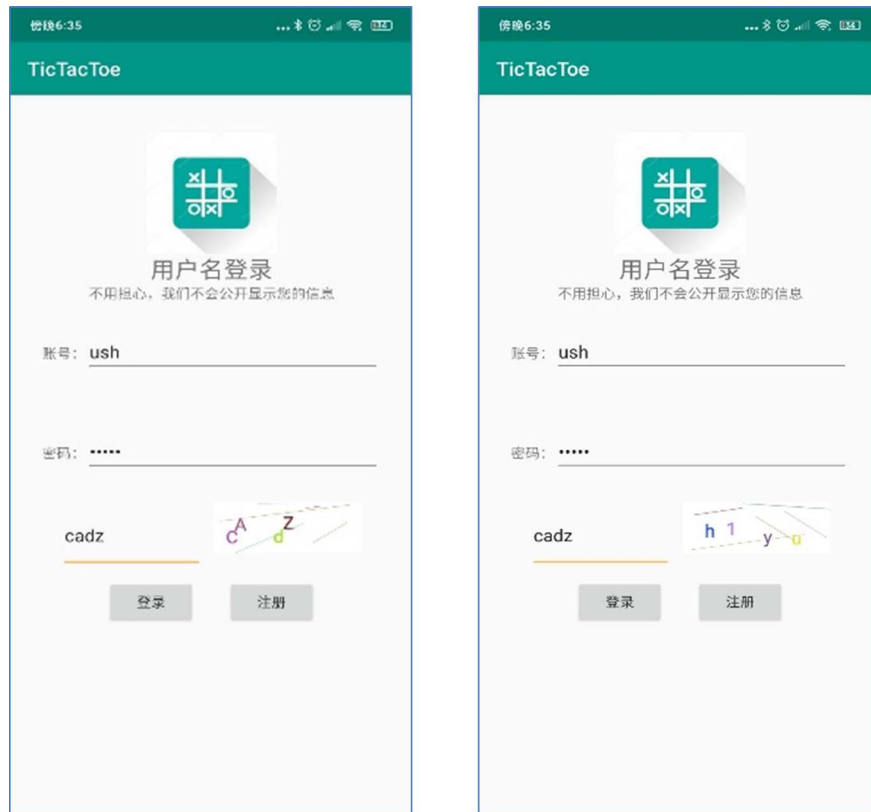


Figure 3. Verification code refresh in APP  
图 3. APP 中的验证码刷新

通过图 3 可以看出, 验证码刷新之后, 就生成了新的验证码。主要是通过设置图片点击事件, 调用算法封装好了的类, 重新生成一个图片验证码并在画布上显示。

#### 4.2. 对比分析

目前市场上应用软件广泛采用的验证码是图片验证码以及滑块验证码, 它们的好处在于用户便于识别, 易操作, 本文基于安卓 SDK 的图片验证码同样具有此优点。

与已有文献相比, 文献[5]和文献[7]采用基于图像处理的图片验证码, 设计思想较为复杂; 文献[9]采用基于常识库随机选取, 部分常识对用户有一定的难度以及机器对常识是易识别的。验证码的主要功能就是加了一道防护, 为的就是机器不能轻松的自动识别。本文基于安卓 SDK 的图片验证码的设计思想设计清晰, 采用 Android SDK 的 API, 在 APP 设计中无需引入第三方插件, 实现起来较为方便。在注册登录之前进行验证码验证提高了安全性, 而且本文的图片验证码是数字和字母组成的, 对用户来说更方便操作。

#### 5. 总结

如今图片验证码已经在各大应用上普遍使用, 本文提出的这款图片验证码算法思想简单易懂, 并尤为高效。在对账号密码验证之前核实验证码, 减轻了对服务器的不断攻击。它的可用性和安全性准则较为可靠, 防止了非法分子的程序自动攻击和人为窃取信息, 结果表明本文的图片验证码在区别用户与机器上是十分有效的。

---

## 参考文献

- [1] 胡健, 柳青, 王海林. 验证码安全与验证码绕过技术[J]. 计算机应用, 2016, 36(S1): 37-41+57.
- [2] 王斌君, 吉增瑞. 信息安全技术体系研究[J]. 计算机应用, 2009, 29(S1): 59-62.
- [3] 姚晨光. 计算机网络与信息安全问题对策[J]. 计算机与网络, 2019, 45(20): 54.
- [4] 赵晓宇, 刘婷婷. 大数据时代的网络信息安全问题研究[J]. 计算机与网络, 2019, 45(11): 47.
- [5] 陈以山. 基于字符的图片验证码识别算法的设计与实现[J]. 电脑知识与技术, 2017, 13(1): 190-192.
- [6] 李友宏. Java Web 应用中加模图片验证码的设计实现[J]. 中国管理信息化, 2018, 21(19): 155-157.
- [7] 曹放. 基于视觉感知的图像验证码设计与研究[D]: [硕士学位论文]. 西安: 西安电子科技大学, 2017.
- [8] 陈义中, 李松林. 一种注册登录系统图片验证码设计与实现[J]. 软件导刊, 2016, 15(7): 82-85.
- [9] 何春辉. 一种基于常识的图片验证码设计方法[J]. 盐城工学院学报, 2018, 31(4): 42-44.