

信息安全工程课题研究

|  |  |
| --- | --- |
| 题目： | Slide CPATCHA |

|  |  |
| --- | --- |
| 成 员： | 龙东恒 许佳 谢贤彬 宋文浩 |
| 院 系： | 软件与微电子学院 |
| 导师姓名： | 孙惠平 |

二〇一七年 十月

# **目录**

[**目录** 1](#_Toc497142067)

[**一、选题背景** 2](#_Toc497142068)

[**二、** **产品现状** 3](#_Toc497142069)

[2.1 验证码发展史 3](#_Toc497142070)

[2.2 无知识型验证码的三大核心优势 3](#_Toc497142071)

[2.3 目前实现滑动认证的一些机制 4](#_Toc497142072)

[2.3.1 现阶段使用状况 4](#_Toc497142073)

[三、研究现状 6](#_Toc497142074)

[**四、主要思路** 8](#_Toc497142075)

[4.1 验证码认证基本流程 8](#_Toc497142076)

[4.2方案设计思路 8](#_Toc497142077)

[4.2.1 方案一 8](#_Toc497142078)

[4.2.2 方案二 9](#_Toc497142079)

[4.2.3方案三 9](#_Toc497142080)

[4.2.4 方案四：最终方案 10](#_Toc497142081)

[4.2.5 四种方案的比较 10](#_Toc497142082)

[4.3 思路整理与问题分析 10](#_Toc497142083)

[五、项目分工 12](#_Toc497142084)

# **一、选题背景**

随着互联网的发展和演进，企业和公司愈来愈重视公布在公众视野中的用户数据，用户搜索的结果列表、用户关注的内容、甚至是用户随手一点的链接，看似无关重要的数据在企业看来当数据量足够大时都将是一块未加开发的宝矿，但是这些数据很容易被业内业外人士通过爬虫等办法获取，以造成隐形的损失。所以，保护企业对外公布的数据成为一项重要任务，其中最需要解决的就是，如何制止机器人盗取数据的现象，而一种流行且高效的办法是：CPATCHA，用以区分人类和机器的验证码。

CPATCHA之所以能够在保护数据中其关键性作用，在于它的核心设计思想——“Easy for human, Hard for Machine”，人类能够轻易识别而机器无法轻易破解。遵照着核心设计思想，验证码在其发展过程中，出现了图片数字识别、多点图片选择、滑动验证码等阶段，并且每一阶段在市场中都流行过较长一段时间。尽管验证码类型多样，但是却很难抵御攻击者各式各样的破解方法，也正因此，验证码技术才会越来越成熟，激励着相关研究人员的研究热情。本项目选取近来噪声大起的滑块验证码作为主题，并对其进行相关论文文献阅读和研究。

# **产品现状**

## 验证码发展史

在计算机系统区分人类与机器行为的发展进程中，验证码作为一项极其关键的技术，共有三段不同的发展阶段，分别为标准验证码、创新验证码以及无知识型验证码三大阶段。其每一阶段的验证码在该阶段中都有相对应的产品诞生，尽管风靡彼时，但各验证码都遭受到攻击者的攻击，并将其逐一突破，这不说明验证码不实用，反而是证实“因为有验证码，人类才要去破解”的论断。对于每一阶段的发展特点，将在下面进行简要描述。

第一阶段为标准验证码。这一代验证码是生活中常见的图形验证码、语音验证码，基于机器难以处理复杂的计算机视觉及语音识别问题，而人类却可以轻松的识别来区分人类及机器。这一代验证码初步利用了人类知识容易解答，而计算机难以解答的机制进行人机判断。

第二阶段为创新验证码。第二代验证码是基于第一代验证码的核心思想（通过人类知识可以解答，而计算机难以解答的问题进行人机判断）而产生的创新的交互优化型验证码。第二代验证码基于第一代验证码的核心原理－－“人机之间知识的差异”，拓展出大量创新型验证码。

第三阶段为无知识型验证码。第三代验证码最大的特点是不再基于知识进行人机判断，而是基于人类固有的生物特征以及操作的环境信息综合决策，来判断是人类还是机器。无知识型验证码最大特点即无需人类思考，从而不会打断用户操作，进而提供更好的用户体验。最著名的例子就是近来格外火热的滑块验证码。

## 无知识型验证码的三大核心优势

正是由于无知识型验证码基于“无需人类思考”的特点，其拥有三大核心优势：

* 更佳用户体验：无知识型验证码针对大多数的用户能够无需思考，直接通过。不存在业务和流程的打断，体验流畅，对用户体验的提升毋庸质疑。
* 更强风险识别：因为随着机器学习的发展让机器掌握人类具有的知识也不再是难点，无知识型验证码不再基于知识来挑战机器，而是基于人类的固有行为特征以及操作的环境信息综合进行风控决策，攻击者难以批量的模拟出可以欺骗风控引擎的正常人类的操作。
* 更高风险拦截：普通的验证码基于知识对机器发起挑战，无法做到对机器进行阻断。因为知识的挑战还需要兼顾人类的体验，机器通过的概率只能做到无限的降低而无法消除。而无知识型验证码基于后端的风控决策，可以对不同风险的操作提出更高难度的验证码乃至阻断，有更大空间对风险进行消除和拦截。

## 2.3 目前实现滑动认证的一些机制

滑动认证原理是通过采集用户的操作数据，环境数据等等，非常多的数据，通过一个算法加密得到字符串，然后提交到服务器分析，服务器有一个判定标准，对数据进行简单的分析就知道是不是人工在操作。

### 2.3.1 现阶段使用状况

现阶段在国内比较火，国外多采用图片点击认证。目前国内比较流行的滑动验证有以下几项：

* 极验验证：现在极验验证码已经更新到了 3.0 版本，截至 2017 年 7 月全球已有十六万家企业正在使用极验，每天服务响应超过四亿次，广泛应用于直播视频、金融服务、电子商务、游戏娱乐、政府企业等各大类型网站。极验验证码增加了机器轨迹识别，匀速移动、随机速度移动等方法都是不行的，只有完全模拟人的移动轨迹才可以通过验证。
* 阿里巴巴：淘宝的UA也是从网页加载好了以后就一直记录用户的操作数据，破解的难度在于JavaScript逆向解密上，当然目前也有很多滑动验证码融合了图形图像的元素，在这个过程中浏览器记录下了你的鼠标移动轨迹，再跟图片验证码的结果相比较，图形的位置不一样，所以你鼠标拖动的长度自然也不一样，这样就无法固定一个操作数据了，即使你要伪造，你也得根据图片的结果来，破解它的难度仍然不在于图片有多难识别，难度仍然在于JavaScript太难解密了，但还是存在破解的方案。只是阿里通过不停的修改高难度加密JavaScript所以保证安全。
* 网易易盾验证码：易盾验证码抛弃了传统字符型验证码展示-填写字符-比对答案的流程，采用验证码展示-采集用户行为-分析用户行为流程，用户只需要产生指定的行为轨迹，不需要键盘手动输入，极大优化了传统验证码用户体验不佳的问题；验证码后台针对用户产生的行为轨迹数据进行机器学习建模，结合访问频率、地理位置、历史记录等多个维度信息，快速、准确的返回人机判定结果，优化了传统验证码基于单一字符维度容易被暴力破解的问题。后台会检测破解攻击，如果有破解的情况，会自动切换到更高难度的点选式验证码；但还是较为容易被破解，所以除此之外网易云还提供终极验证上行短信验证，确保验证码的安全性。

# 三、研究现状

滑块验证码技术中主要的技术难点集中于拼图的选取和用户行为轨迹的判断。拼图（Puzzle）考量其拼图形状、拼图大小、图片背景中拼图选取的位置以及拼图挖取后空白处颜色填充等指标。通过改变以上指标，能够对其难度进行适当的调整，其关系如下：

* 拼图形状：拼图边数越多、拼图边界轨迹越复杂，则对于攻击者所使用的边界检测算法的要求更苛刻，难度系数成正比增加。
* 拼图大小：大小的设定主要涉及拼图所覆盖底层图片的面积大小，过大或过小的拼图面积都不利于提高区分度，但是对于适中面积的选取，目前还没有相关文献对其进行过讨论研究。
* 图片背景中拼图选取的位置：主要考虑背景图片的色彩分布、色度、亮度等图片属性，当拼图选取于颜色较为复杂（满足特定的数学条件，例如DCT）的位置，攻击者若试图使用暴力破解，其所得结果未必是合理合法结果。
* 拼图挖取后空白处颜色填充：主要影响攻击者试图利用背景图片与拼图块空白处之间的颜色边界的区分，当边界颜色差别大时，攻击者可以直接利用边界检测算法获得拼图正确位置，而当边界颜色差别小时，边界检测算法则很可能失效。

以上四点是文献中反复提及的拼图要素，攻击者常根据上述要素进行攻击。但除此之外，若攻击者已经识别出拼图正确位置，则他需要通过一定手段和策略命令机器人模仿人类移动拼图的方式，将拼图拖拽至指定位置。此过程中，拼图移动的最小单位像素距离、拼图移动过程速度变化以及拼图移动方向轨迹将会影响判断结果，影响关系如下：

* 移动最小单位像素距离：影响最终的拼图与空白处的重合度（或精确度），人类操作过程很难做到百分百重合率，而机器人却能轻而易举地做到；
* 移动过程速度变化：人类在进行拼图拖拽时，其速度存在生物学上的统计规律，攻击者需要掌握并了解后台行为算法检测才能够攻破
* 移动过程方向轨迹：同速度变化一样，人类在进行拼图拖拽时其拖拽行为是多变、多方向的，非单一行为，同样存在生物学上的统计规律对于攻击者而言也是一种挑战。

综合以上对于拼图和用户行为的影响要素，滑块验证码能够做到非常优秀的区分度。但是在实际应用过程中，若公司企业并不是格外看重其暴露的数据时，他们会在高精度区分性和实用性上进行选择，而结果显而易见，即更看重使用的便捷性、用户友好度，一切以用户出发。

在研究的过程中，选择适当牺牲友好度进而换取更优秀的区分效果，虽然离产品化仍有很长距离，但仍旧有机会一试。

# **四、主要思路**

## 4.1 验证码认证基本流程

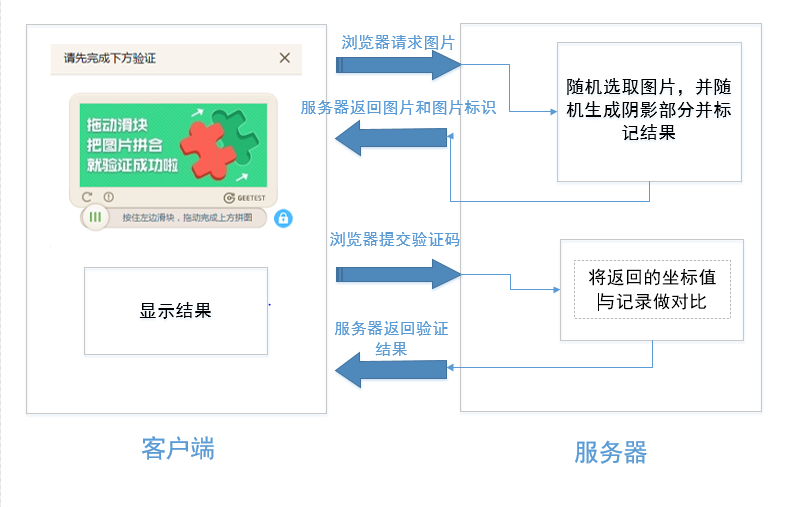


图1：使用servlet实现验证码基本流程

## 4.2方案设计思路

### 4.2.1 方案一

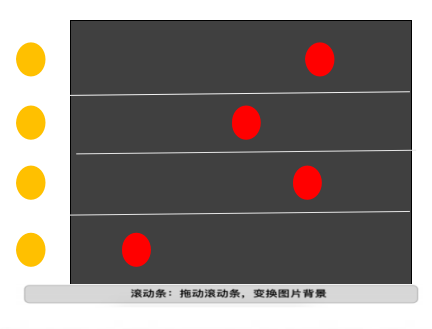


图2：方案1示意图

此方案将图片划分为四行滑动模块，用户必须每行都按照从上到下的顺序，且每行都验证正确才能验证成功。注意：每行验证后并不立即判断是否正确，直到所有行均验证完成，才返回服务器数据，并判断是否验证成功。

### 4.2.2 方案二

此方案将图片划分为两行滑动模块，每行滑动模块形状可能正好对应相应的阴影区形状，也可能正好相反。用户只有先点击switch按钮，选择正确的模块并将两个模块均正确的拼到阴影区，才能够验证成功。

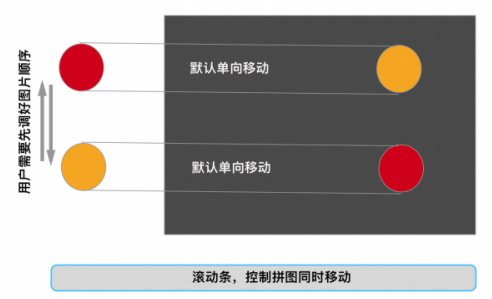


图3：方案二示意图

### 4.2.3方案三

此方案一次验证包含四张图片，整体划分为四块，但只有一个滑动区与一个滑动块，用户必须将这一个滑动块拼到正确的图片与形状区域才能验证成功。

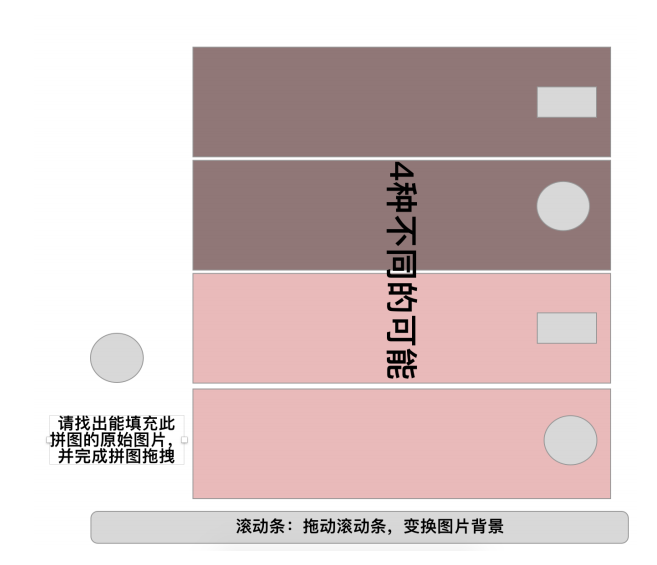


图4：方案三示意图

### 4.2.4 方案四：最终方案

此方案的图片中两个滑动阴影区，验证时必须先点击左下角的旋转按钮选择正确的滑动模块，将滑动模块拼到正确的阴影区，验证成功。



图5：方案四示意图

### 4.2.5 四种方案的比较

|  |  |  |
| --- | --- | --- |
| 方案 | 优点 | 缺陷 |
| 方案一 | 增强了对蛮力破解的抵御强度 | ➀对用户更不友好  ➁没有增强对其他攻击方式抵御能力 |
| 方案二 | ➀对用户来说更加快捷简单  ➁比常用的滑动验证更安全 | 在四种方案中安全性最差 |
| 方案三 | 安全性高，面面兼顾，能够有效地抵御各种攻击方法，尤其是蛮力攻击 | ➀设计太复杂，难度较高  ➁用户不友好，此方案平均每个用户都要多次点击转换按钮并分辨四张图，用户难以区分且增加了验证时间 |
| 方案四 | 既有方案三中用阴影区形状来辨别，又有判断拼图块区域，更安全更不易被破解 | ➀对一些图片，随机选取的阴影区用户会不好判断拼图块位置  ➁对图片有一定的要求 |

表1：四种方案的优缺点比较

仔细考虑过后，我们组选择方案四作为这次的最终方案。

## 4.3 思路整理与问题分析

针对我们选择方案，我们提出了以下问题和难点，有待分析和解决。

① 如何在图片中显示一个不规则阴影区

② 如何随机显示阴影区的位置

③ 如何实现滑动

④ 如何进行坐标验证

⑤ 前后关联如何实现

⑥ 是否应该有时间限制

⑦ 是否需要错误次数限制

⑧ 拼图块旋转如何实现

# 五、项目分工

项目的最终成品是一个自定义的滑块验证码系统，以实现“展示-拖拽-认证-反馈”简单验证码认证流程为目的。其中，在实现过程中采用前后端分离的模式，实施微服务架构，即前端业务由专门Frontend服务器负责，并以特定的Endpoint对外暴露服务（前端主要负责接收来自Backend服务器的反馈信息）；后台业务由专门负责同数据库进行交互的Backend服务器完成（负责存储验证码信息、用户行为信息以及验证结果），并对外暴露数据接口。Backend与Frontend通过互相调用各自的接口完成业务实施，这类模式具有低耦合高内聚特点，双方能够通过微小的改动，完成整体业务的部署。



图X 组内项目分工思维导图

基于以上的部署结构，分工简要分为前端和后台两部分工作，具体细节有：

1. 基于 JavaScript 的图片拖动、滚动条带动拼图背景（或拼图块）变化；
2. 基于 JavaScript 的鼠标行为记录（点击、拖拽等记录时间、轨迹）；
3. 基于 JavaScript 的表单提交；
4. 基于 JavaScript 的 Frontend 前端展示页面；
5. 基于 任意语言 的拼图图片生成（生成拼图块算法、去除拼图块生成空白处算法、保存操作后的拼图及拼图块）；
6. 确定前端展示图片大小，拼图块位移数据格式；
7. 人工生成拼图块及拼图背景（labeled），需要使用PS及进行拼图块最终位置记录（Excel）；
8. 基于 MySQL 的拼图库建立与维护；
9. 基于 任意语言 的 Backend 后台服务，包括响应请求、接受请求、判断是否成功（涉及判断算法）;
10. 确定前端后台数据交互格式;
11. 检验与测试。

分工采用自由获得机制，组员根据自己能力和喜好领取相关的任务。