

浙江千麦司法鉴定中心

司法鉴定意见书



千麦司法鉴定
Chain Forensic Science

声 明

一、司法鉴定机构和司法鉴定人按照法律、法规和规章的规定，按照鉴定的科学规律和技术操作规范，依法独立、客观、公正进行鉴定，并出具鉴定意见，不受任何个人或者组织的非法干预。

二、委托人应向本机构提供真实、完整、充分、符合鉴定要求的鉴定材料，提供虚假鉴定材料导致鉴定意见错误的，本机构不承担任何责任。

三、本鉴定意见仅对本次委托人提供的鉴定材料负责。

四、司法鉴定意见书是否作为定案或者认定事实的根据，取决于办案机关的审查判断，司法鉴定机构和司法鉴定人无权干涉。

五、如对本鉴定书内容有任何异议或者疑问请尽早与本机构取得联系。根据司法部《司法鉴定程序通则》四十一条之规定，本机构在必要时，可以对鉴定文书进行补正或出具补正书。

六、使用司法鉴定意见书，应当保持其完整性和严肃性。未经本鉴定机构的书面同意，任何单位或个人不得部分复印本鉴定意见书（全部复印除外）。

七、鉴定意见属于鉴定人的专业意见，当事人对鉴定意见有异议，应当通过庭审质证或者申请重新鉴定、补充鉴定等方式解决。

网址：www.zjcfs.com 电话：0571-85852181 传真：0571-85852191
投诉电话：0571-85256592（杭州市司法局）
地址：杭州市余杭区余杭街道华一路2号4号楼

浙江千麦司法鉴定中心 司法鉴定意见书

浙千司鉴中心[2019]电鉴字第333号



一、基本情况

委托人：浙江省杭州市国立公证处

委托事项：对送检笔记本电脑中摇号软件的程序功能进行鉴定；对“2019年9月28日”摇号软件产生的数据库数据进行分析。

委托日期：2019年9月29日

受理日期：2019年9月29日

鉴定材料：

1. 数据光盘1张（标签“登记名单”，以下称检材一）；
2. 数据光盘1张（标签“摇号软件”，以下称检材二）；
3. ThinkPad 笔记本电脑1台（P/N: SL10M60885，以下称检材三）。

鉴定日期：2019年9月29日至2019年10月11日

鉴定地点：浙江千麦司法鉴定中心电子数据鉴定实验室

二、基本案情

据委托书记载：浙江省杭州市国立公证处因在2019年9月28日下午五点的摇号结果中发现大量连号的情况，为查明事实，现委托本中心对送检笔记本电脑中摇号软件的程序功能进行鉴定；对“2019年9月28日”摇号软件产生的数据库数据进行分析鉴定。

三、资料摘要

无。

四、鉴定过程

依据 GA/T 976-2012《电子数据法庭科学鉴定通用办法》、GB/T 29362-2012《电子物证数据搜索检验规程》、SF/Z JD0403004-2018《软件功能鉴定技术规范》和 GA/T 1480-2018《法庭科学计算机操作系统仿真检验技术规范》，使用电子物证工作站、电子数据仿真取证系统、取证大师专业版 V6.1.65366、AccessData FTK Imager 3.1.2.0、jd-gui 1.4.0 反编译工具、Microsoft Access 2007 对检材进行数据提取固定分析。

1. 检材检验

启动杀毒软件对电子物证工作站系统进行杀毒，确保工作站系统安全。

1.1 送检检材一为贴有“登记名单”标签的数据光盘 1 张，检查外观完好，在其外部贴上唯一性标识“DZ190929JC01”，并进行拍照固定，见附图 1 所示；将检材一通过只读光驱连接至电子物证工作站；使用“AccessData FTK Imager”软件对检材制作镜像，文件保存为“20190929 杭州市国立公证处光盘（DZ190929JC01）.iso”，计算其 SHA256 值为“3eecd8970f949718b6450de27f392212263b7f384fe51686593f30ae689f0670”，见附图 2-3 所示。

1.2 送检检材二为贴有“摇号软件”标签的数据光盘 1 张，检查外观完好，在其外部贴上唯一性标识“DZ190929JC02”，并进行拍照固定，见附图 4 所示；将检材二通过只读光驱连接至电子物证工作站；使用“AccessData FTK Imager”软件对检材制作镜像，文件保存为“20190929 杭州市国立公证处光盘（DZ190929JC02）.iso”，并计算其 SHA256 值为

“d6cf71659ef64b3d5c9d2981d713e0bf9de8dcfb7bafd582b8e855e436736f7”，见附图 5-6 所示。

1.3 送检检材三为“ThinkPad”笔记本电脑一台，检查外观完好，在其外部贴上唯一性标识“DZ190929JC03”，并进行拍照固定；取出其中的硬盘，硬盘品牌为“Intel”，标称容量为 256GB，在其外部贴上唯一性标识“DZ190929JC03-1”并进行拍照固定，见附图 7-9 所示；将检材硬盘通过只读接口连接至电子物证工作站；使用“AccessData FTK Imager”软件制作镜像，文件保存为“DZ190929JC03-1.dd4.001”，并计算其 SHA256 值为“51f683d7c22d4f800210d4cc5d8a3f3f1b19c3e721ded26e2d9837b9254efd8f”，见附图 10-12 所示。

2. 导出数据

将检材一中的文件导出，共有 1 个文件，名为“未隐藏.xls”，大小为“488KB”，内容为本次参加摇号家庭的名单，计算其 SHA256 值为“9682fb5b788d9a8304417847b05cf2cd70868735041b2842809ba2b0f6f41817”，见附图 13 所示；

将检材二中的文件导出，共有一个名为“摇号程序文件”的文件夹内含 25 个文件，计算 25 个文件的 SHA256 值见下表所示。

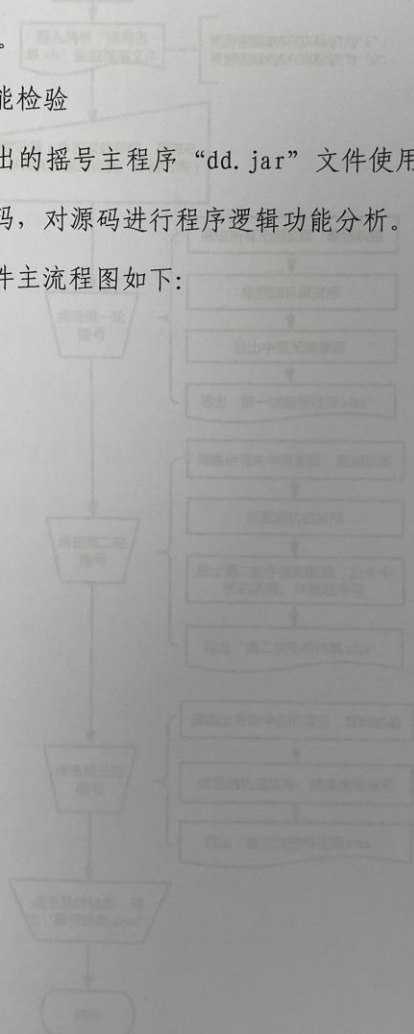
文件名	SHA256 值
initname.txt	e1b9b6df5d3ee5d7700d52a07b6be564339ed5f601e794624fff4ed85d28e6dc
initname.txt.bak	be5beedb38d0ad0eb3a4ca07f48d550885c0772600a61b18f67ac063b225c095
_ .jpg	0e16482d8430a5ea7c8d51feb11c4100fa71fab9e2e387b103ef36fc023313f8
_ .jpg	9a3ac0f424eefa9b67aa89564746f0d2fe4e90739b49eb91a80c398b2a94957e
11.jpg	b646df8252e048f859b292f9c87cf7282414f0618e8eb17f96c0a65e65cdca2b

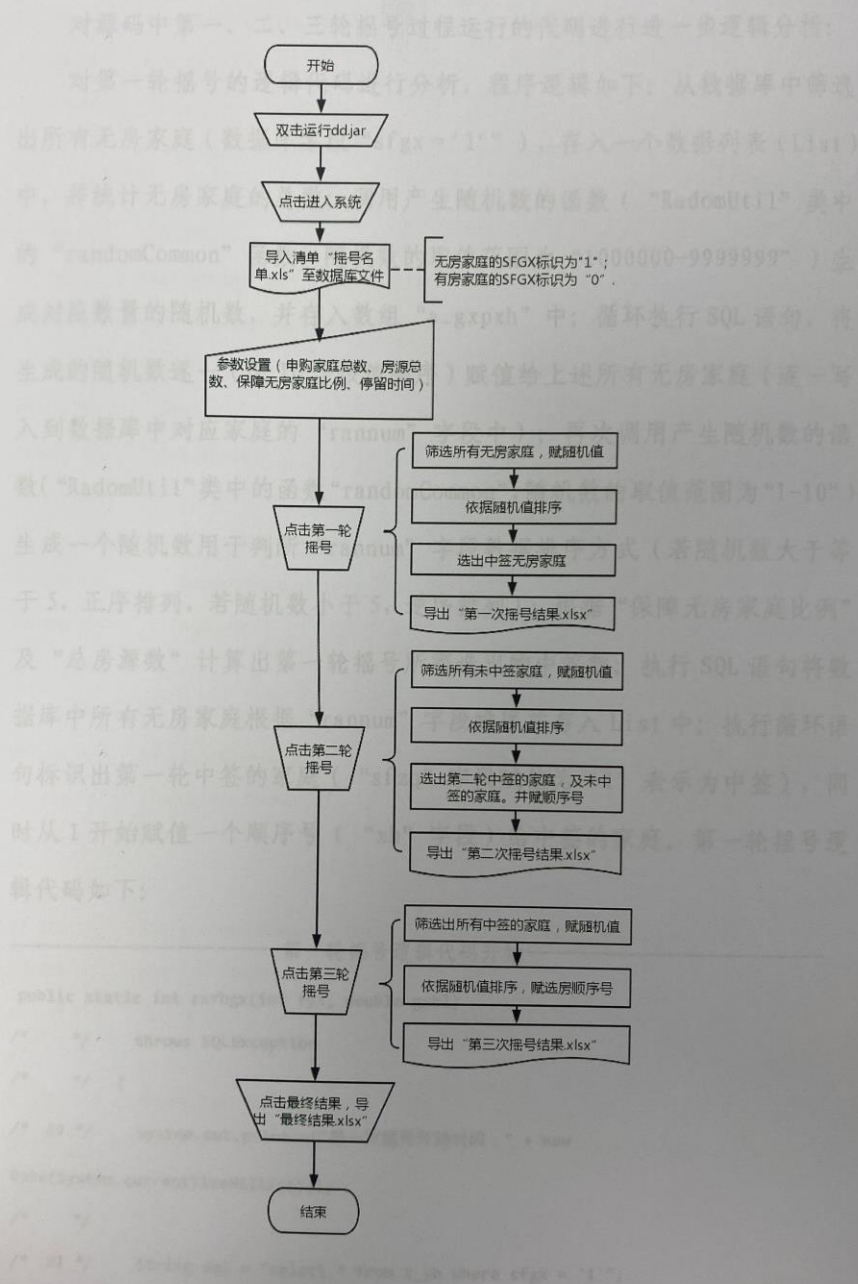
btninit. jpg	bdc3e3588b3ce41167022b46b37ce360b76ccf6bcd3bc29ffbbd949a54e6eaa
btnjixu. jpg	ad9dc01afb76d27771aac36c55641ddb8918dbbda6a13ebce39cf30221abf11d
btnset. jpg	453c9f77c28440ee71b5b9f3fb1891238b63ce57c61b597672a7956a6930aca7
btnyaohao. jpg	f7749c93ca6986a5eb439748c3ce0986a9b2b584bb483c4f7268ed400df03d54
btnzantin. jpg	9083b473486760c55421b9133750d61e53b8b7eb4cd97788b61e59cb4f5c4cb1
frmyaohao. jpg	e8ffdbeed63f55554a67ddc50b60f7c4e1146ac18616ec44681a5e2bb84a75b
imglist. jpg	64c2795ef5316b5a19d3a6e9dcfa5a5b002b32b94a4aabdb3b75422be318290f
yhjg. jpg	ad374d1387b3e2c395386dd74e9c182da0945683a2bb981430aa523f143e7a02
yhjg2. jpg	dc72e12472e9b6738873e361abc6c05a021e10e19665d1ece5c46398a472ecee
XF. mdb	5c3a9338c9cc79d47d0d484a2271c2d05b739ccb70c9aa2e4fa1f00d3b97dfd3
申购登记表模板. xls	d3491bc489378d82f7e8bc8b68f42afd303a8405725e1e1eb5a7ef9b0d293726
申购登记表模板_30. xls	5679da469e4c81ffee8153608a7e691debeb32fc6adbbbe80351889b7020fcc1
申购登记表模板_2000. xls	fa7c29cecccd5c6cccf8ceb4ce6c0442f465b664f304c509f2fad7576830547fb
申 购 登 记 表 模 板 _30000. xls	d3c109a369cd450474208237667b9b4389ca27285ed8224ea853308f4eee785c
申 购 登 记 表 模 板 _65535. xls	da3609b1cfc3625531ab614e0e3d2302681a4d38829d1e502b23b30016ebbdd2
AccessDatabaseEngine_X6 4. exe	0c41d59a27984fd3032149a2f065841de9609dd3f756b3c3ba4836f02e7ac7c0
jdk-6u45-windows-x64. ex e	345059d5bc64275c1d8fdc03625d69c16d0c8730be1c152247f5f96d00b21b00
dd. jar	44feba1977e9ba018715fce9bbd90de2a870de72f7f7ae8d6483b98a65adable
脚本. bat	70565d356e0db2bdd243b6cb5180fe286a1ecbf38f8dec2e32676aec12d07b7
使用说明文档. docx	ea2e55df7d3f0b898335857b2a99accebd80420e701269243677b0b8d39bd5c8

根据委托鉴定要求，将检材三硬盘镜像中本次摇号相关的“摇号程序文件”、“CS”、“JG”文件夹和“未隐藏.xls”文件导出至路径“D:\2019年9月29日浙江省杭州市国立公证处\DZ190929JC03-1\应用程序数据固定”下，见附图14所示。

将检材三中导出的摇号主程序“dd.jar”文件使用“jd-gui”软件进行反编译获取其源码,对源码进行程序逻辑功能分析。

分析整理出软件主流程图如下:





对源码中第一、二、三轮摇号过程运行的代码进行进一步逻辑分析:

对第一轮摇号的逻辑代码进行分析, 程序逻辑如下: 从数据库中筛选出所有无房家庭(数据库字段“sfzx='1'”), 存入一个数据列表(List)中, 并统计无房家庭的总数; 调用产生随机数的函数(“RadomUtil”类中的“randomCommon”函数, 随机数的取值范围为“1000000-9999999”)生成对应数量的随机数, 并存入数组“a_gxpxh”中; 循环执行 SQL 语句, 将生成的随机数逐一(按 ID 字段值顺序)赋值给上述所有无房家庭(逐一写入到数据库中对对应家庭的“rannum”字段中); 再次调用产生随机数的函数(“RadomUtil”类中的函数“randomCommon”, 随机数的取值范围为“1-10”)生成一个随机数用于判断“rannum”字段数据排序方式(若随机数大于等于 5, 正序排列, 若随机数小于 5, 逆序排列); 根据“保障无房家庭比例”及“总房源数”计算出第一轮摇号所需选出的中签数; 执行 SQL 语句将数据库中所有无房家庭根据“rannum”字段排序并存入 List 中; 执行循环语句标识出第一轮中签的家庭(“sfzq”字段赋值为‘1’表示为中签), 同时从 1 开始赋值一个顺序号(“xh”字段)给中签的家庭。第一轮摇号逻辑代码如下:

```
-----第一轮摇号逻辑代码开始-----
public static int zxYhgx(int fys, double gxbl)
/*      */ throws SQLException
/*      */ {
/* 89 */ System.out.println("第一次摇号开始时间:" + new
Date(System.currentTimeMillis()));
/*      */
/* 91 */ String sql = "select * from t_yh where sfzx = '1'";
```

```

/* 92 */ List<Map<String, Object>> l = AccessDb.query(sql);
/* 93 */ int zs = l.size();
/* 94 */ sql = "select * from t_yh where rannum = '1' order by rannum";
/* 95 */ int[] a_gpxh = RadomUtil.randomCommon(1000000, 9999999, zs);
/* */
/* 97 */ int i = 0;
/* 98 */ List<String> o = new ArrayList();
/* 99 */ for (Map<String, Object> m : l) {
/* 100 */     sql = "update t_yh set rannum = " + a_gpxh[i] + " where id = " + m.get("id");
/* 101 */     i++;
/* 102 */     o.add(sql);
/* 103 */     i++;
/* 104 */     if (o.size() == 5000) {
/* 105 */         AccessDb.excute(o);
/* 106 */         o.clear();
/* 107 */     }
/* */ }
/* 109 */ if (o.size() > 0) {
/* 110 */     AccessDb.excute(o);
/* 111 */ }
/* */ }

/* 113 */ int[] randomCommon = RadomUtil.randomCommon(1, 10, 1);
/* 114 */ int r = randomCommon[0];
/* 115 */ String pxh = r >= 5 ? " asc " : " desc ";
/* */
/* 117 */ BigDecimal b_gxfys =
BigDecimal.valueOf(fys).multiply(BigDecimal.valueOf(gxbl)).divide(BigDecimal.ONE,
java.math.RoundingMode.DOWN);
/* */
/* 119 */ int i_gxfys = b_gxfys.intValue();

```

```

/* 120 */ return i;
/* 121 */ i = 1;
/* 122 */ sql = "select * from t_yh where sfzx = '1' order by rannum " + pxh;
/* 123 */ l = AccessDb.query(sql);
/* 124 */
/* 125 */ o.clear();
/* 126 */ for (Map<String, Object> m : l) {
/* 127 */     if (i > i_gxfys) {
/* 128 */         break;
/* 129 */     }
/* 130 */     sql = "update t_yh set xh = " + i + ", sfzq = '1' where id = " + m.get("id");
/* 131 */     i++;
/* 132 */     o.add(sql);
/* 133 */     if (o.size() == 5000) {
/* 134 */         AccessDb.execute(o);
/* 135 */         o.clear();
/* 136 */     }
/* 137 */ }
/* 138 */ if (o.size() > 0) {
/* 139 */     AccessDb.execute(o);
/* 140 */ }
/* 141 */
/* 142 */ o = null;
/* 143 */ l = null;
/* 144 */
/* 145 */ System.out.println("第一次摇号结束时间：" + new
Date(System.currentTimeMillis()));
/* 146 */

```



```

/* 147 */    return i;
/* 148 */    List<Map<String, Object>> l = AccessDB.query(sql);
/* 149 */    Map<String, Object> map = (Map)l.get(0);
/* 150 */    int i = Integer.parseInt(map.get("count").toString());
-----第一轮摇号逻辑代码结束-----

```

对第二轮摇号的逻辑代码进行分析，程序逻辑如下：从数据库中统计出第一轮已中签的家庭数量；筛选所有未中签家庭（除第一轮已中签家庭之外的所有家庭），存入一个数据列表（List）中并统计总数；调用产生随机数的函数（“RadomUtil”类中的“randomCommon”函数，随机数的取值范围为“1000000-9999999”）生成对应数量的随机数，并存入数组“a_gxpxh”中；执行 SQL 循环语句，将生成的随机数逐一（按 ID 字段值顺序）赋值给上述所有未中签家庭（逐一写入到数据库中对应家庭的“rannum”字段中）；再次调用产生随机数的函数（“RadomUtil”类中的函数“randomCommon”，随机数的取值范围为“1-10”）生成一个随机数用于判断“rannum”字段数据排序方式（若随机数大于等于 5，正序排列，若随机数小于 5，逆序排列）；执行 SQL 语句将数据库中所有未中签家庭按“rannum”字段排序并存入 List 中；标识出第二轮中签的家庭（“sfzq”字段赋值为‘1’表示中签），同时从“第一轮中签数+1”开始赋值一个顺序号（“xh”字段）给本轮中签家庭和未中签的家庭（“sfzq”字段赋值为‘0’表示未中签）。第二轮摇号逻辑代码如下：

```

-----第二轮摇号逻辑代码开始-----
/* 151 */    public static int zxYhsy(int fys)
/* 152 */    throws SQLException
/* 153 */    {
/* 154 */
/* 155 */
/* 156 */
/* 157 */
/* 158 */
/* 159 */    System.out.println("第二次摇号开始时间：" + new
Date(System.currentTimeMillis()));
/* 160 */
/* 161 */

```



```

/* 161 */      String sql = "select count(*) as n from t_yh where sfzq = '1'";
/* 162 */      List<Map<String, Object>> l = AccessDb.query(sql);
/* 163 */      Map<String, Object> map = (Map)l.get(0);
/* 164 */      int j = Integer.parseInt(map.get("n").toString());
/* 165 */      AccessDb.excute(o);
/* 166 */      sql = "select * from t_yh where sfzq = '0' ";
/* 167 */      l = AccessDb.query(sql);
/* 168 */      int zs = l.size();
/* 169 */      l = null;
/* 170 */      int[] a_gpxh = RadomUtil.randomCommon(1000000, 9999999, zs);
/* 171 */
/* 172 */      int i = 0;
/* 173 */      List<String> o = new ArrayList();
/* 174 */      for (Map<String, Object> m : l) {
/* 175 */          sql = "update t_yh set rannum = " + a_gpxh[i] + " where id = " + m.get("id");
/* 176 */          i++;
/* 177 */          o.add(sql);
/* 178 */          if (o.size() == 5000) {
/* 179 */              AccessDb.excute(o);
/* 180 */              o.clear();
/* 181 */          }
/* 182 */      }
/* 183 */      if (o.size() > 0) {
/* 184 */          AccessDb.excute(o);
/* 185 */      }
/* 186 */
/* 187 */      int[] randomCommon = RadomUtil.randomCommon(1, 10, 1);
/* 188 */      int r = randomCommon[0];
/* 189 */      String pxh = r >= 5 ? " asc " : " desc ";
/* 190 */
/* 191 */      o.clear();
/* 192 */      sql = "select * from t_yh where sfzq = '0' order by rannum " + pxh;
/* 193 */      l = AccessDb.query(sql);
/* 194 */
/* 195 */      for (Map<String, Object> m : l) {
/* 196 */          j++;
/* 197 */          if (j > fys) {
/* 198 */              sql = "update t_yh set xh = " + j + ", sfzq = '0' where id = " + m.get("id");
/* 199 */          } else {
/* 200 */              sql = "update t_yh set xh = " + j + ", sfzq = '1' where id = " + m.get("id");
/* 201 */          }
/* 202 */          o.add(sql);
/* 203 */          if (o.size() == 5000) {
/* 204 */              AccessDb.excute(o);

```

```

/* 205 */      o.clear();
/* */      }
/* */      }
/* 208 */      if (o.size() > 0) {
/* 209 */          AccessDb.excute(o);
/* */      }
/* */      System.out.println("第三次摇号开始时间：" + new
/* 212 */      o = null;
/* 213 */      l = null;
/* */      String sql = "select * from t_xh where sfzq = '1' ";
/* 215 */      System.out.println("第二次摇号结束时间：" + new
Date(System.currentTimeMillis()));
/* */      int[] a_gxpxh = RadomUtil.randomCommon(1000000, 9999999, 11);
/* 217 */      return j;
/* */      }

```

-----第二轮摇号逻辑代码结束-----

对第三轮摇号的逻辑代码进行分析，程序逻辑如下：从数据库中选出第一、二轮所有已中签的家庭（“sfzq”字段赋值为‘1’表示中签），存入一个数据列表（List）中，并统计中签家庭总数；调用产生随机数的函数（“RadomUtil”类中的“randomCommon”函数，随机数的取值范围为“1000000-9999999”）生成对应数量的随机数，并存入数组“a_gxpxh”中；执行 SQL 循环语句，将生成的随机数逐一（按 ID 字段值顺序）赋值给上述中签家庭（逐一写入到数据库中对应家庭的“rannum”字段中）；再次调用产生随机数的函数（“RadomUtil”类中的函数“randomCommon”，随机数的取值范围为“1-10”）生成一个随机数用于判断“rannum”字段数据排序方式（若随机数大于等于 5，正序排列，若随机数小于 5，逆序排列）；执行 SQL 语句对所有中签家庭按“rannum”字段进行排序；执行循环语句按排序逐一给所有中签家庭从 1 开始赋值顺序号（逐一写入到数据库中对家庭的“xh”字段中）。第三轮摇号逻辑代码如下：

-----第三轮摇号逻辑代码开始-----

```

/* */ public static int px()
/* */ throws SQLException
/* */ { AccessDb.excute(o);

/* 229 */ System.out.println("第三次摇号开始时间：" + new
Date(System.currentTimeMillis()));
/* */ l = null;
/* 231 */ String sql = "select * from t_yh where sfzq = '1' ";
/* 232 */ List<Map<String, Object>> l = AccessDb.query(sql);
/* 233 */ int zs = l.size();
/* */ System.out.println("第三次摇号开始时间：" + new
Date(System.currentTimeMillis()));
/* 235 */ int[] a_gpxh = RadomUtil.randomCommon(1000000, 9999999, zs);
/* */
/* 237 */ int i = 0;
/* 238 */ List<String> o = new ArrayList();
/* 239 */ for (Map<String, Object> m : l) {
/* 240 */ sql = "update t_yh set rannum = " + a_gpxh[i] + " where id = " + m.get("id");
/* 241 */ i++;
/* 242 */ o.add(sql);
/* 243 */ if (o.size() == 5000) {
/* 244 */ AccessDb.excute(o);
/* 245 */ o.clear();
/* */ }
/* */ }
/* 248 */ if (o.size() > 0) {
/* 249 */ AccessDb.excute(o);
/* */ }
/* */
/* 252 */ int[] randomCommon = RadomUtil.randomCommon(1, 10, 1);
/* 253 */ int r = randomCommon[0];
/* 254 */ String pxh = r >= 5 ? " asc " : " desc ";
/* */
/* 256 */ o.clear();
/* 257 */ sql = "select * from t_yh where sfzq = '1' order by rannum " + pxh;
/* 258 */ l = AccessDb.query(sql);
/* 259 */ i = 1;
/* 260 */ for (Map<String, Object> m : l) {
/* 261 */ sql = "update t_yh set xh = " + i + ", sfzq = '1' where id = " + m.get("id");
/* 262 */ i++;
/* 263 */ o.add(sql);
/* 264 */ if (o.size() == 5000) {
/* 265 */ AccessDb.excute(o);

```



```
/* 266 */      o.clear();
/* */      }
/* */      }
/* 269 */      if (o.size() > 0) {
/* 270 */          AccessDb.excute(o);
/* */      }
/* */
/* 273 */      o = null;
/* 274 */      l = null;
/* */

/* 276 */      System.out.println("第三次摇号结束时间：" + new
Date(System.currentTimeMillis()));
/* 277 */      return i;
/* */ }
```

-----第三轮摇号逻辑代码结束-----

进行完上述三轮摇号后已经生成相应的最终数据结果，再将最终结果从数据库中全部导出至 Excel 表中并生成“最终结果.xlsx”文件。

分析所有源码，未发现有人为控制摇号结果的代码语句。

4. 模拟仿真检验

使用“电子数据仿真取证系统”对检材三镜像文件进行系统仿真加载，对摇号软件进行运行。进入其操作系统后，在系统桌面发现存放摇号软件的“摇号程序文件”文件夹和存放本次参与摇号名单的“未隐藏.xls”文件；见附图 15 所示。

通过双击运行“摇号程序文件”文件夹中“dd.jar”打开摇号软件，点击“进入系统”后，将摇号名单“未隐藏.xls”文件导入软件，点击“参数设置”并输入“本期可售房源总数：92”、“保障无房家庭比例（%）：30.5”后，点击确定；见附图 16-19 所示。此时将软件运行过程中“C:\CS”路径下用于存放数据的“XF.mdb”数据库文件另存至虚拟机系统桌面“各阶段数据\导入名单”文件夹内。打开数据库文件，发现与导入表格中的数

据对应一致，对应列名和字段名、数据库新增列名见下表。

未隐藏.xls 表列名	XF.mdb 数据库字段名	备注
购房登记号	DJBH	/
购房人姓名	XM	/
购房人证件号码	ZJHM	/
是否无房家庭	SFGX	无房家庭为“1”，有房家庭为“0”
查档编号	CDBH	/
其他购房人及家庭成员姓名	QTCYXM	/
其他购房人及家庭成员证件号码	QTCYSFZH	/
/	ID	数据库主键 ID
/	RANNUM	随机数
/	XH	顺序号
/	SFZQ	是否中签
/	CXY	/

点击“第一轮摇号”，待页面上第一轮摇号结果显示停止后，此时将软件运行过程中“C:\CS”路径下用于存放数据的“XF.mdb”数据库文件另存至虚拟机系统桌面“各阶段数据\第一轮摇号”文件夹内，见附图 20 所示。打开数据库文件，发现所有无房家庭的“RANNUM”字段均被赋予随机数；根据“RANNUM”字段值排序后，发现前 28 个家庭被标识为已中签（“SFZQ”字段为‘1’）且被赋予序号（“XH”字段分别为 1 至 28 号）。

点击“第二轮摇号”，待页面上第二轮摇号结果显示停止后，此时将软件运行过程中“C:\CS”路径下用于存放数据的“XF.mdb”数据库文件另存至虚拟机系统桌面“各阶段数据\第二轮摇号”文件夹内，见附图 21 所示。打开数据库文件，发现所有家庭的“RANNUM”字段均被赋予随机数；剔除第一轮已中签的家庭后，根据“RANNUM”字段值排序，发现前 64 个家

庭被标识为已中签（“SFZQ”字段为‘1’）且被赋予序号（“XH”字段分别为 29 至 92 号）；剩余家庭被标识为未中签（“SFZQ”字段为‘0’）且被赋予序号（“XH”字段分别为 93 至 2164 号）。

点击“第三轮摇号”，待页面上第三轮摇号结果显示停止后，此时将软件运行过程中“C:\CS”路径下用于存放数据的“XF.mdb”数据库文件另存至虚拟机系统桌面“各阶段数据\第三轮摇号”文件夹内，见附图 22 所示。打开数据库文件，发现所有已中签家庭（包括第一轮、第二轮中签家庭）的“RANNU”值被重新赋予随机数，并被重新赋予序号（“XH”字段分别为 1 至 92），所有未中签家庭的“XH”字段与第二轮该字段中的数据均对应一致。

点击“最终结果”，待页面上最终摇号结果显示停止后，此时将软件运行过程中“C:\JG”路径下用于存放数据的“XF.mdb”数据库文件另存至虚拟机系统桌面“各阶段数据\最终结果”文件夹内，见附图 23 所示。打开数据库文件，发现数据与第三轮摇号结束的数据均对应一致。

5. 数据库数据检验

对“2019 年 9 月 28 日”摇号产生的“XF.mdb”数据库文件进行分析，共 2164 条数据记录，其中 494 个无房家庭，1670 个有房家庭；发现部分数据记录中“RANNU”（“RANNU”字段存放用于排序的随机数）字段存在数据为“NULL”（空）的情况，根据摇号程序逻辑正常运行的情况下，在三轮摇号结束后，“RANNU”字段内的值不应存在数据为“NULL”（空）的情况。故对“RANNU”字段存在为“NULL”（空）情况出现的程序逻辑位置进一步进行分析：

第一轮摇号逻辑应为筛选出本次参与的 494 个无房家庭，并对这些家庭的“RANNUM”字段逐一（按 ID 字段值顺序）赋随机数；根据“RANNUM”进行排序，取前 28 个作为本轮中签的家庭，并逐一赋予一个顺序号（从 1 至 28）。故通过数据库筛选出无房家庭（即数据库字段“SFGX = '1'”）的数据记录共 494 条，逐一检查发现“RANNUM”字段均不为“NULL”（空）。

第二轮摇号逻辑应为筛选所有未中签的 2136 个家庭（除第一轮已中签 28 个家庭之外的所有家庭），并对这些家庭的“RANNUM”字段逐一（按 ID 字段值顺序）赋随机数；根据“RANNUM”进行排序，给前 64 个作为本轮中签的家庭逐一赋予一个顺序号（从 29 至 92）；给之后 2072 个未中签的家庭逐一赋予一个顺序号（从 93 至 2164）。故按 ID 字段值顺序逐一检查所有家庭的“RANNUM”字段，发现从 ID 字段=“1485345”之后（包含本数）且“XH”字段值大于 92 的有房家庭数据记录中，均显示“RANNUM”字段为“NULL”（空）；“XH”字段值小于等于 92 的所有家庭均显示“RANNUM”字段不为“NULL”（空）。一中“未中签.xls”文件对应相同。

第三轮摇号逻辑应为筛选出所有 92 个中签家庭（包括第一轮中签的 28 个家庭、第二轮中签的 64 个家庭），并对这些家庭的“RANNUM”字段逐一（按 ID 字段值顺序）赋随机数；根据“RANNUM”进行排序，逐一给中签家庭赋予一个顺序号（从 1 至 92，逐一写入到数据库中对应家庭的“XH”字段中）。故根据序号顺序排序逐一查看所有家庭的数据记录，发现顺序号为 1 至 92 号家庭的“RANNUM”字段均不为“NULL”（空）；顺序号为 93（包含本数）至 1582 号（包含本数）家庭的“RANNUM”值均为“NULL”（空）；顺序号为 1583 号（包含本数）至 2164 号（包含本数）家庭的“RANNUM”

字段均不为“NULL”（空）。（空）的情况，不符合程序逻辑功能正常运行

对第二轮、第三轮出现的数据“RANNUM”字段存在部分为“NULL”（空）的情况，结合程序的逻辑功能分析发现，在第二轮摇号程序运行过程中，程序按照ID顺序逐一给家庭（除第一轮已中签家庭以外的所有家庭）赋予随机数时，ID字段在“1485345”之后（包含本数）的家庭并未成功赋予随机数（导致“RANNUM”字段为“NULL”（空））；但程序继续按照“RANNUM”进行顺序排序（“RANNUM”字段为“NULL”（空）的家庭在前，“RANNUM”字段为不为“NULL”（空）的家庭在后），并对前64个家庭标记为中签。第三轮摇号程序运行过程中，将第一、二轮已中签的家庭赋予随机数用于确定选房顺序。

五、分析说明

1. 经过比对发现送检检材三中导出的“摇号程序文件”文件夹中的文件与检材二中“摇号程序文件”文件夹中的文件均对应相同；检材三中的“未隐藏.xls”文件与检材一中“未隐藏.xls”文件对应相同。
2. 通过对摇号程序文件进行反编译，分析程序逻辑功能，符合将导入的参与摇号的名单，根据房源数、保障无房家庭比例进行三轮随机摇号产生中签名单和选房顺序的功能需求；未发现可以人为控制摇号结果的代码语句。通过对送检笔记本的系统进行仿真，运行系统内的摇号软件，对“数据导入”、“第一轮摇号”、“第二轮摇号”、“第三轮摇号”、“最终结果”五个阶段产生的数据库文件分别进行导出和分析，各阶段产生的数据与程序逻辑功能相符，未发现数据异常，软件运行过程正常。

3. 发现本次摇号生成的数据库数据部分有房家庭数据记录存在

“RANNU” 字段为 “NULL”（空）的情况，不符合程序逻辑功能正常运行的结果。通过分析 “RANNU” 字段为 “NULL” 值的分布情况并结合源码的逻辑功能，分析第一轮参与摇号的 494 个无房家庭均未出现 “RANNU” 字段为 “NULL”（空）的情况；分析得出部分有房家庭数据记录存在 “RANNU” 字段为 “NULL”（空）情况的原因 6 为第二轮摇号程序运行过程中出现未对部分数据记录的 “RANNU” 字段进行赋随机数值，从而导致第二轮摇号排序过程中，根据 “RANNU” 字段排序时将 “RANNU” 字段为 “NULL” 值的数据记录排序在前，故第二轮中签的家庭均为 “RANNU” 字段为 “NULL” 值的有房家庭，最终出现本次摇号结果的异常情况。

4. 数据固定

将导出的数据及摇号软件反编译后的源码打包压缩为文件 “DZ190333.rar”，并封盘刻录至专用的光盘中，光盘编号为 “浙千司鉴中心 [2019] 电鉴字第 333 号-ZT”。光盘内文件 SHA256 值为 “307ddfbb2a0df0b2608d3a143202d94e5610e123a201a5719be2ae8d3f45d31d7”。

六、鉴定意见

通过对检材进行数据提取固定并对摇号程序功能、数据库数据进行鉴定，结果如下：

1. 摇号软件的程序功能为将导入的参与摇号的名单，根据房源数、保障无房家庭比例进行三轮随机摇号产生中签名单和选房顺序。未发现可以人为控制摇号结果的代码语句。

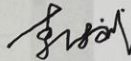
2. “2019 年 9 月 28 日” 摇号的第一轮摇号数据未发现异常情况，第二

轮摇号存在部分数据未被赋予随机值的异常情况。

七、附件

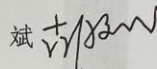
1. 附件图片共 9 页，内含 23 张图；
2. 编号为“浙千司鉴中心[2019]电鉴字第 333 号-ZT”数据光盘 1 份，内含“DZ190333.rar”文件。

司法鉴定人：李佳斌



执业证号：330114070049

司法鉴定人：胡斌



执业证号：330114070008

二〇一九年十月十二日



图 2 检材一数据文件

图 3 检材一数据光盘

附件图片

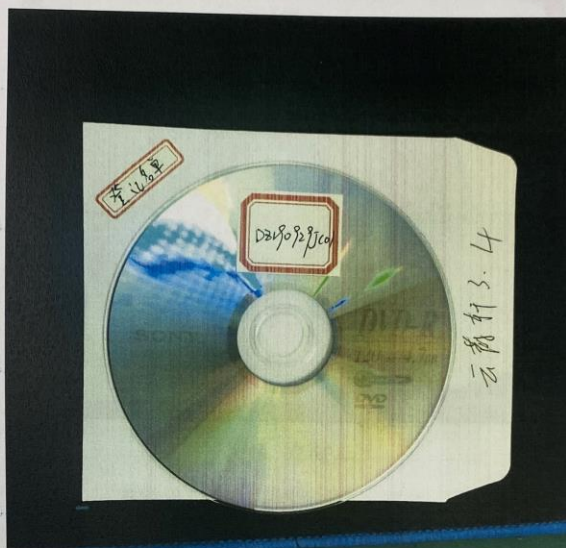


图1 检材一



图2 检材一镜像文件

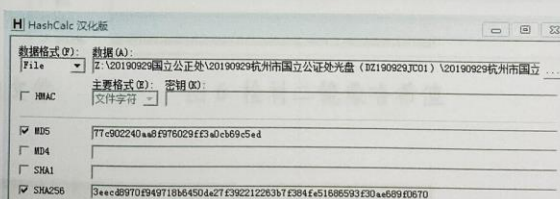


图3 检材一镜像哈希值

图7 检材三正面

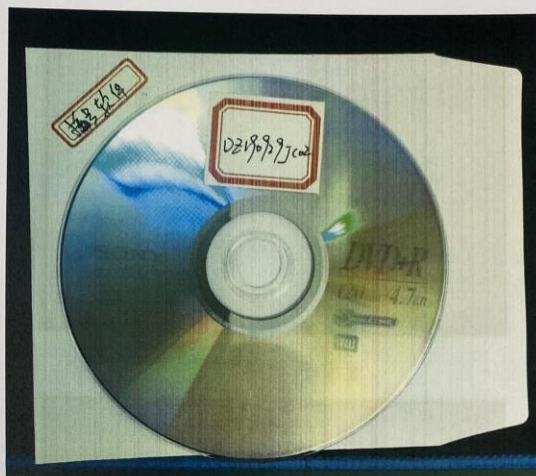


图4 检材二



图5 检材二镜像文件

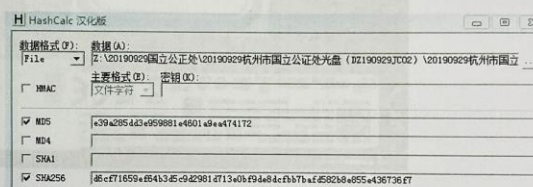


图6 检材二镜像哈希值

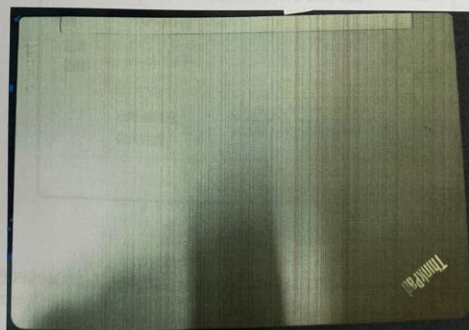


图7 检材三正面



图8 检材三背面

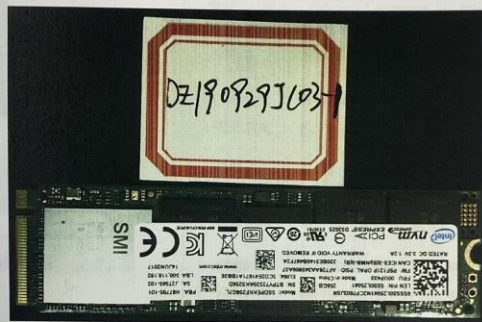


图9 检材三硬盘

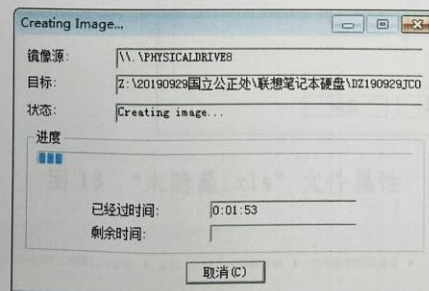


图10 制作镜像

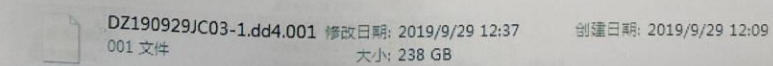


图11 检材三镜像文件

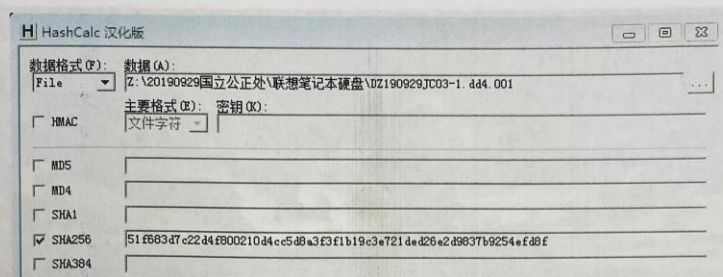


图 12 检材三镜像哈希值

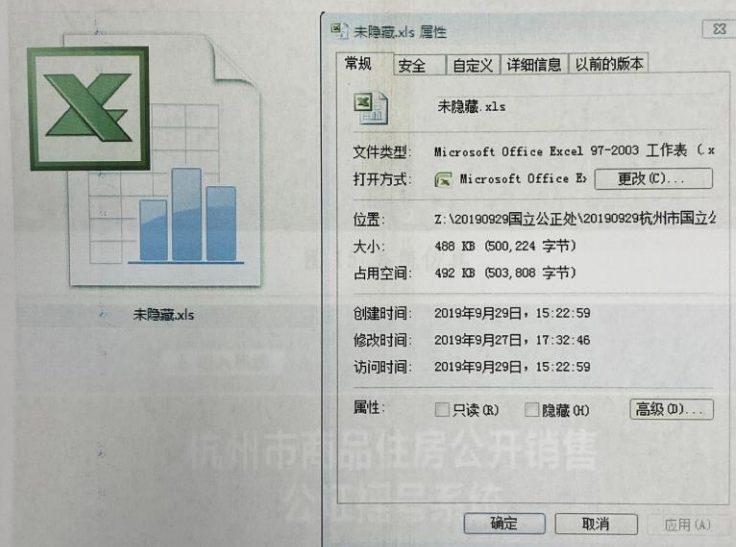


图 13 “未隐藏.xls”文件属性



图 14 导出文件



图 15 系统仿真

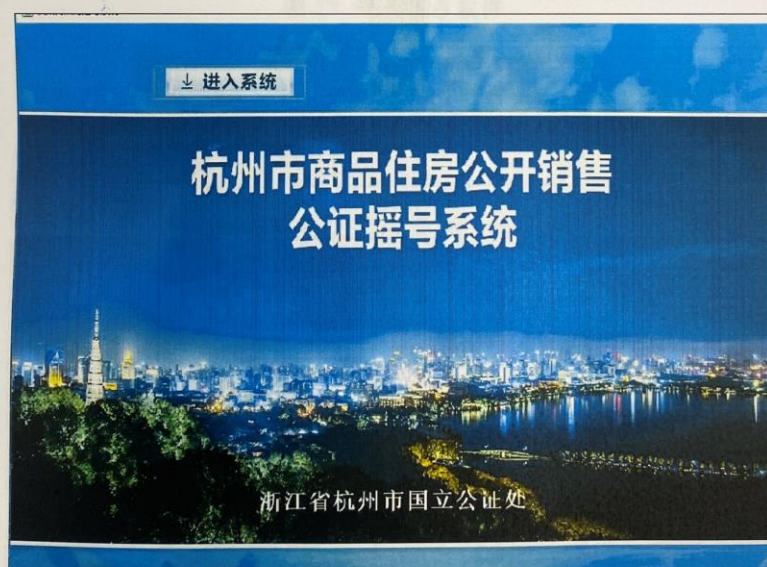


图 16 摇号软件界面

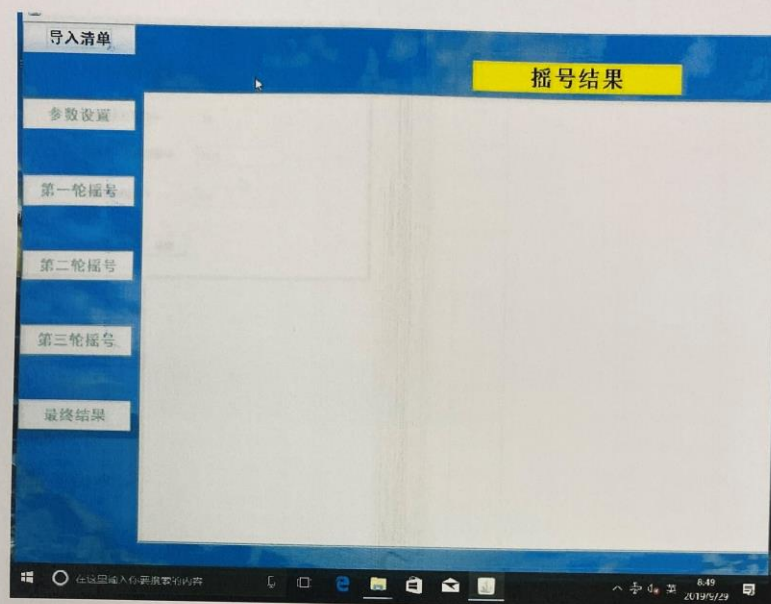


图 17 进入摇号软件

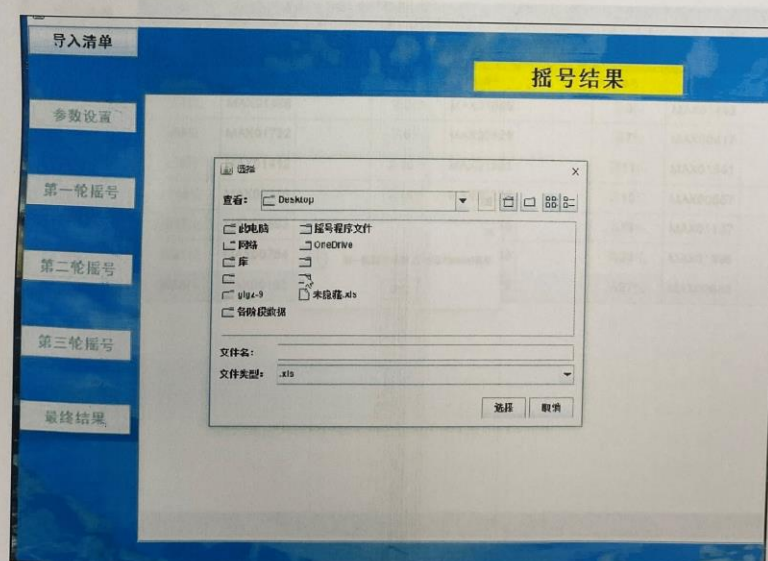


图 18 导入名单

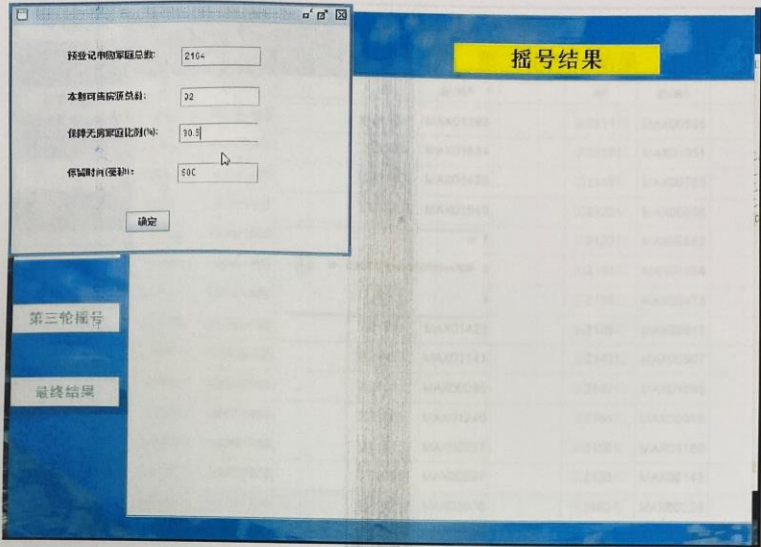


图 19 设置参数



图 20 第一轮摇号

图 21 第二轮摇号

图 22 第三轮摇号

导入清单		最终摇号结果					
参数设置	序号	登记编号	序号	登记编号	序号	登记编号	
第一轮摇号	第2109位	MAXC2059	第2110位	MAXD1392	第2111位	MAXD0565	
	第2113位	MAXC1861	第2114位	MAXD1634	第2115位	MAXD1951	
	第2117位	MAXC0335	第2118位	MAXD1438	第2119位	MAXD0765	
第二轮摇号	第2121位	MAXC1311	第2122位	MAXD1940	第2123位	MAXD0006	
	第2125位	MAXC1809			第2127位	MAXD0352	
	第2129位	MAXC1802			第2131位	MAXD1494	
第三轮摇号	第2133位	MAXC1485			第2135位	MAXD0473	
	第2137位	MAXC0155	第2138位	MAXD1422	第2139位	MAXD0811	
	第2141位	MAXC0105	第2142位	MAXD1141	第2143位	MAXD0067	
最终结果	第2145位	MAXC0195	第2146位	MAXD0085	第2147位	MAXD1090	
	第2149位	MAXC1469	第2150位	MAXD1240	第2151位	MAXD0069	
	第2153位	MAXC1755	第2154位	MAXD0727	第2155位	MAXD1180	
	第2157位	MAXC0322	第2158位	MAXD0687	第2159位	MAXD2145	
	第2161位	MAXC1330	第2162位	MAXD2000	第2163位	MAXD0229	

图 23 最终结果

数据载体粘贴单

案件号：浙千司鉴中心[2019]电鉴字第333号

载体编号：浙千司鉴中心[2019]电鉴字第333号-ZT

载体内文件：“DZ190333.rar”文件，文件SHA256值为
“307ddfbb2a0df0b2608d3a143202d94e5610e123a201a5719be2ae8d3f45d31d7”。



备注：切勿剪切载体内的文件，请复制后校验哈希值再解压。

执业机构: 浙江千麦
司法鉴定中心

专业技术职称: _____

行业执业资格: _____

执业范围: 声像资料鉴定
(电子数据鉴定)

有效期: 2019年07月18日至2024年07月17日
首次获准登记日期: 2019年07月18日
颁证机关: 浙江省司法厅
颁证日期: 2019年07月18日

姓名: 李佳斌

性别: 男

执业证号: 330114070049

身份证号码: 330324199201122819



执业机构: 浙江千麦
司法鉴定中心

专业技术职称: 高级工程师

行业执业资格: _____

执业范围: 痕迹鉴定
声像资料鉴定
(电子数据鉴定)

有效期: 2019年07月25日至2024年07月24日
首次获准登记日期: 2014年07月25日
颁证机关: 浙江省司法厅
颁证日期: 2019年07月25日

姓名: 胡斌

性别: 男

执业证号: 330114070008

身份证号码: 330802196206074030





中华人民共和国
司法鉴定许可证 (正本)

机构名称：浙江千麦司法鉴定中心
统一社会信用代码：34330000031361621XG
法定代表人：陈娅妮
首次获准登记日期：2014年7月25日
机构负责人：潘豪杰
机构住所：杭州市余杭区余杭街道华一路2号4号楼
业务范围：法医学病理鉴定，法医临床鉴定，法医物证鉴定，法医毒物鉴定，文书鉴定，痕迹鉴定，声像资料鉴定（电子数据鉴定）

颁证机关：浙江省司法厅

有效期限：2019年07月25日至2024年07月24日
颁证日期：2019年07月25日

中华人民共和国司法部监制

浙江千麦司法鉴定中心服务项目

- 1、**法医临床鉴定**：伤残等级评定、伤病关系鉴定、损伤程度鉴定（轻、重伤鉴定）、致伤物和致伤方式推断、三期评估（误工、护理、营养时限评定）、活体年龄鉴定、护理依赖程度评定、工伤鉴定、男子性功能鉴定、视觉功能鉴定、听觉功能鉴定、医疗损害鉴定、医疗费合理性评定；
- 2、**法医病理鉴定**：死亡原因鉴定、死亡方式鉴定、死亡时间鉴定、生前伤与死后伤鉴定、硅藻检验、病理组织学诊断、伤病关系参与度鉴定、医疗纠纷尸体检验、致伤（死）物推断；
- 3、**法医物证鉴定**：亲权鉴定、个体识别、种属认定、有无人血迹、有无人精斑；
- 4、**法医毒物鉴定**：乙醇定性定量分析、生物检材中常见毒品定性分析、体外疑似毒品定性定量分析、常见毒（药）物筛选定性分析、碳氧血红蛋白饱和度测定（一氧化碳中毒）、氰化物定性定量分析；
- 5、**文书鉴定**：笔迹鉴定，印章印文鉴定，朱墨时序鉴定，打印、传真、复印等印刷文件鉴定，变造文件鉴定，污损文件的显现、整复、辨读等，书写工具及其他工具形成的文字压痕及其它压印痕迹的显现、辨读、鉴别等；
- 6、**痕迹鉴定**：车辆痕迹鉴定、车速鉴定、指印检验鉴定、足迹检验鉴定、工具痕迹鉴定、整体分离痕迹鉴定；
- 7、**声像资料（电子数据）鉴定**：手机、硬盘等存储介质数据恢复、提取、固定，数据库等电子数据分析，网络数据获取、固定，软件功能鉴定，软件相似性鉴定，电子邮件分析，密码破解，电子数据真实性（完整性）鉴定等。