

解决一下视频中提出的问题

1 订阅外包服务

优先按照接口文档来，前端的「协商密钥」输入框可以直接删掉了

2 认证与密钥协商接口

这个接口没有你理解的那么复杂，这里的协商和我们登录的协商是两个东西，这里只是为了展示我们协商的这个功能，所以在这儿生成一套新的随机数，然后三个模块的内容都是统一一套的（即，**通过按钮点击获取到的内容是一个文件中的不同部分，而不是每个部分都单独重新生成**）。

这个接口有小改动，具体可以看下面的第二点

3 溯源部分

这个部分请直接看第三点的详细说明

部分没在文档中体现出来的内容：

1. 跨域访问控制部分（基于现有需要改进的地方）

这个部分需要改进的地方主要是在于文件下载的两个接口，也就是**域内、域外文件下载**两个接口

1 需要前端维护一个文件列表

和师姐协商过后，我们就不多开接口来获取新的文件列表了，需要前端简单维护一个文件列表（本来应该是区块链来维护的，但是实际操作很麻烦所以就直接改成前端了，也就不需要后端再重新写一个新的接口了）

这个文件列表的内容就是用户每一次使用「**文件上传**」之后的文件信息，其中具体包括的内容大致有：“文件名”、“文件所属用户”、“所在域”，**存储方式**就用json就可以，怎么方便后续获取传值怎么来就行。

The screenshot shows a web form titled "文件上传" (File Upload). It has three main input areas: "文件名" (Filename) with the value "file0", "文件内容" (File Content) with the placeholder "This is the content.", and "访问结构" (Access Structure) with the placeholder "((2 AND 4) OR 6)". At the bottom right are "取消" (Cancel) and "上传" (Upload) buttons. A red rectangular box highlights the "文件名" input field. To the right of this box, there is a red text annotation: "根据该文件名及当前的用户（即文件所属用户），以及当前所属的域名生成一个文件列表" (Generate a file list based on the filename and the current user (i.e., the user to whom the file belongs), and the current domain).

这里提供一个师姐那边之前考虑的存储方式：

```
{ "Bdomain": { "BdomainList": [ "Bobfile0" ] },  
  
  "Adomain": { "AdomainList": [ "Alicefile0" ] } }  
// 这里不单独直接使用fileList来作为一个json文件是为了如果之后需要存储其他信息，可以使用这个json一并解决，就不需要多个文件了  
// 「Adomain」 and 「Bdomain」：表示所在域  
// 「AdomainList」 and 「BdomainList」：表示为文件列表  
「Alicefile0」 and 「Bobfile0」：文件命名方式统一为：文件所属用户名+文件名
```

⚠ 注意：

- 文件名为：文件所属用户 + 文件名
- 接口传值的时候，需要提供的为「当前登录用户名」和「文件名」，不需要再提供「文件所属用户名」了

2 文件下载的弹窗简化

在域内访问、域外访问的情况中，都会有文件下载的弹窗，这个时候需要前端直接判断「当前登录用户的所属域」，再根据具体的域内访问还是域外访问做区分，在前端直接提供可下载的文件列表。

前端界面变化：把输入用户名、文件名的两个文本框入口都统一一个下拉列表就可以了，因为现在的文件名均包含两个信息：文件所属用户名、文件名

3 跨域访问服务弹窗简化

在这个接口中，前端通过本地维护的文件列表，提供两个信息给弹窗：「目标域列表」+「文件列表」

前端界面变化：

- 「目标域」放在最上方，做成一个下拉列表，展示可以选择的域名（不包括自己的）
- 根据「目标域」的选择结果，返回不同的「文件列表」，然后通过下拉列表展示结果
- 本地域不再需要输入，根据前端暂存的所属域的值直接从接口发送即可

跨域访问服务 本地域：直接从前端保存值返回，不用输入
文件名：通过下拉列表获取
目标域：做成一个下拉列表，放在「文件名」
上面，这样做出之后，根据所选域来展示
这个用户可以访问到的其他域的文件

请输入文件名、本地域、目标域

| | |
|-----|---------|
| 文件名 | Alice0 |
| 本地域 | Adomain |
| 目标域 | Bdomain |

取消 下载

2. 认证与密钥协商部分（基于现有需要改进的地方）

这个部分的内容最简单，其实就只需要把几个按钮的接口统一一下，然后直接一口气展示最终的几个结果即可

简单来说：三个「值获取」按钮改成一个，然后通过这个按钮点击之后就能够直接展示所有的相关密钥协商的结果信息

额外需求（评估下如果不好实现就算了），在前端展示的时候，因为还是分三个模块的内容，所以我想要最终效果是点击按钮之后，第一个模块、第二个模块、第三个模块的内容都是逐渐出现的。效果类似于：第一个模块的内容先一口气展示出来，等1s再展示第二个模块的内容，再等1s展示第三个模块的内容

| | | | |
|---|-------|-----|-------|
| 用户A的公钥 761D584F...87 | 第0s展示 | 获取值 | |
| 用户B的公钥 904BC779...93 | | | |
| 用户B的会话密钥 3EEDAEC5C80A3FEE60B14C91C7703C4DF0BFF1C0 | | | |
| 用户B的生成协商函数值 0x3A 0xFC 0xCF 0x24 0x5A 0xB5 0x53 0x19 | | | |
| | | | |
| 用户B存储值 | 第1s展示 | 获取值 | |
| 用户B存储值sessionID AB1 | | | |
| 用户B存储值hash 4532fb07a2e2d5c3a700f422c7f68cde2934d57250c257b6f523849d0a47628d | 去掉 | 获取值 | |
| | | | |
| 用户A获取并计算 | | | 第2s展示 |
| 用户A的会话密钥 | 获取值 | | |
| 用户A获取存储值sessionID | | | |
| 用户A获取存储值hash | | | |
| 用户A计算存储值 | | | |

3. 异常流量检测与溯源部分（对需要做的内容做一下说明）

溯源部分有一些背后的逻辑处理需要做

1 简单解释问题

因为溯源有非常多的流量包信息，然后这些信息最初的接口发送形式是：「后端直接把所有数据都发送给前端，等前端分对每类数据都分别统计处理之后，再在页面上展示出所需要的信息」。

但是我觉得这样做对于前端的要求太多，并且效率不够高，所以让后端对部分数据做了处理之后，再返回数据给前端。

2 目前的实现方法

具体来说，你可以参考一下需求文档里面的接口说明，应该写清楚了现在接口的实现方法。这里我简单写一下我的理解：

1. 总体的实现逻辑

- 每一次点击到“异常流量监测与溯源”这个模块之后，会触发三个接口：「/detect/result」、「/detect/resultlist」、「/detect/all」
- 这三个接口的用途在「系统需求文档」中有图片说明，直接照着文档写即可；但是all接口有额外说明内容

对于all接口，需要额外说明：

- 接口返回的内容为「本次点击」到「上次点击」之间产生的数据统计量
- 前端需要对这些统计量做加法处理，将后一次和前一次的加起来得到最终的结果