PULA 课件 Flash加密方案

V1.1 TIYI 2014

保密文档

# 目的

为保证由PULA制作，基于Flash技术的课件不被轻易盗用，因此本方案提出了多个技术手段，涵盖加密解密、密钥管理等内容，避免课件大规模外流引起的数字资产损失。

# 定义

运行容器：呼叫Flash课件进行运行的win32桌面客户端；有传入加密值的义务。

密钥：容器与Flash课件通信的加密基础值。本方案中为16个字节的英数组合，例如c13124f32117de81。每个课件均持有一份密钥，通常与其他课件不重复。

# 实现要求

## 运行容器

运行容器在实例中为开课客户端。通过主系统的课程部署，以联网同步的方式获得了运行Flash课件的对应的加密密钥，并在本地加密存储。

运行时，通过对应加载Flash课件，并通过定时传入参数的方式实现加密值传递，并提供回报（callback）接口，由Flash呼叫回调回报运行状态。

### 定时传值

运行容器根据课件对应的密钥，根据当前客户端系统时间，将加密值传入至Flash的参数接口；时间间隔为60秒

加密公式为：

加密结果=BASE64(AES( UNIX时间戳 ,密钥))

例如：

1391396469037 密钥为：c13124f32117de81

结果: YfjxUh4eHh5BGzDDHTxyOpp95jTE

由Flash负责解读、校验有效性。

### 回报接口

接收Flash端运行状态回报，状态码和对应说明见附录。

## Flash课件

采用AS3.0实现

留意：

import flash.external.ExternalInterface;

### 密文接收

提供接口接收容器传入的密文，并解密成时间戳。

采用：

ExternalInterface.addCallback ("**pula\_key**", pula\_key);

function pula\_key(s:String)

{

//解密s 并处理

}

### 定时检查

由Flash课件每隔30秒检查一次上一次传入的时间戳。与当前系统时间比对是否时差在±120秒以上，如果超过该值，界面提示无效访问。

如时间戳无效，界面同样提示无效访问。

### 状态回报

当Flash运行状态发生变化时，需通过容器的回报接口，传入加密状态码。

采用：

ExternalInterface.call ("**pula\_state**", **状态码**); //取消str

# AS示例代码

文中附上一份AS实现的AES 加解密代码及接口实现。用以解读AES是否正常、传值是否正常。请参考: flash\_interface.rar

# 辅助工具

## 课件加密校验器

课件加密校验器用以检查Flash课件是否按照指定接口要求制作课件

### 模拟定时传输正确的密文确保运行过程正常，以检测Flash制作是否正常。

### 模拟错误密文，以检测Flash是否检验密文有效性。

### 自动生成一段随机密钥，用以Flash课件制作时使用。

请参考:CallFash-Demo,全部解压后CallFlash.exe

# 附录

## 状态码及说明

|  |  |
| --- | --- |
| 状态码 | 说明 |
| 0 | 正常运行 |
| 1 | 校验密文错误（无法解密） |
| 2 | 无效时间戳（逾期） |
| 99 | 课件执行完毕🡪转至作品提交画面 |