加密服务

1. 加密服务

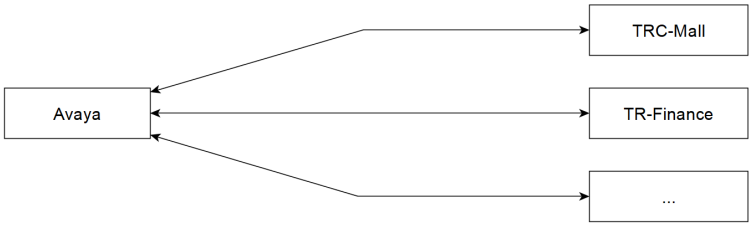
1.1. 背景与需求

公司引入Avaya客服系统，此系统需要调用各业务系统的服务以便于客服在话务过程中执行相关操作， 由于此系统未开放源码，出于安全考虑各业务系统要对返回结果中的敏感字段进行加密，Avaya存储的是加密信息，在前端展现时再执行解密操作。

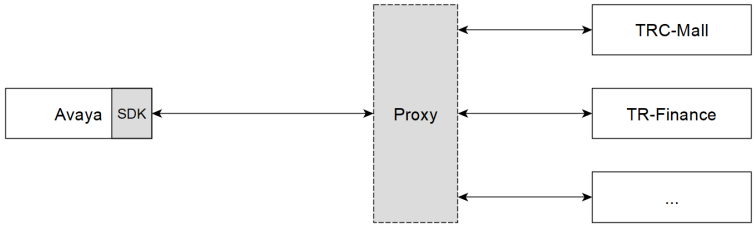
1.2. 实现分析

以上需求可由独立的加解密服务实现，加、解密都由此服务内部实现，不存在密钥下发情况，出于性能考虑可选择对称加密AES实现，密码为MD5(种子+随机码)以减少“撞库”风险。

1.3. 服务迁移



上图是最始的服务调用，没有引入TEval及加密服务。



引入TEval后的调用如上，迁移操作为：

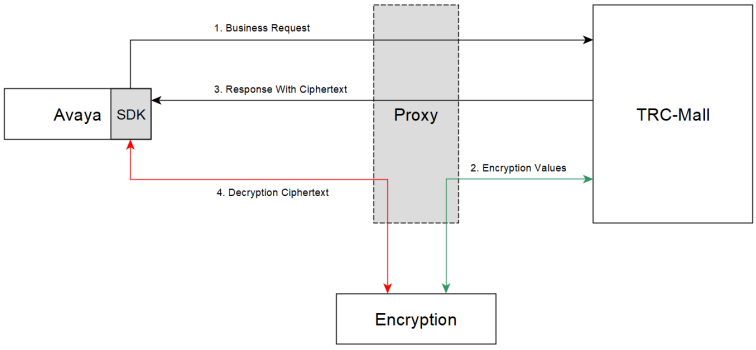
服务提供方（电商、金融等）

- 在TEval中配置并发布服务

服务调用方Avaya

- 在TEval中注册一个账号，获取AccountSecret
- 在TEval中订阅需要的服务
- 在程序中嵌入对应语言及框架版本的SDK
- 添加AccountSecret配置（Classpath/其它指定路径/启动参数等地方）

Proxy对服务提供及调用方都是透明的，两方也不需要修改代码，包含原来请求的URL、参数都不用变更！



引入加密服务后的调用如上，迁移操作为：

服务提供方（电商、金融等）

- 在TEval中订阅加密服务
- 在收到业务请求后返回前自行判断哪些字段要加密，调用TEval上注册的加密服务完成加密并返回，密文格式为：Cipher{<加密后的值,此次加密的随机码>}

服务调用方Avaya

- 在TEval中订阅加密服务
- Avaya在展现时判断值是否是Cipher{...}格式，如果是则需要调用TEval上注册的加密服务完成解密并返回明文

为什么需要显示地调用加解密而不在TEval中做透明实现

1. 出于性能考虑TEval的Proxy不解析HTTP　Body，故无法通过Proxy的插件体系实现
2. TEval目前不考虑支持服务聚合，所以无法将一次请求先路由到提供方后再路由到加密服务
3. Avaya的需求是部分加密（敏感字段），将加解密操作交由使用方更为灵活

1.4. 加密服务接口契约

加密.

```
POST HTTP://<TEval Host>:<TEval Port>/encryption/encrypt

Request Body:
[
   {"key":"<字段名>","value":"<字段值>"},
   ...
]

Response Body:
[
   {"key":"<字段名>","value":"Cipher{<密文>}"},
   ...
]
```

解密.

```
POST HTTP://<TEval Host>:<TEval Port>/encryption/decrypt

Request Body:
[
   {"key":"<字段名>","value":"Cipher{<密文>}"},
   ...
]

Response Body:
[
   {"key":"<字段名>","value":"<字段值>"},
   ...
```

```
]
```