**License使用帮助**

# license库

* 采用rsa加密机制，提高license生成的安全性；
* 生成算法容器的参考码，并用私钥签名生成license；
* license增加时效性，以便用于不同的场景，例如测试，试用；
* license增加请求量上限限制，用于细分算法处理能力；
* license增加版本号，方便之后升级兼容；
* license库的实现方案可以公开，并提供静态库链接。

**库名，例如**

libji\_license.a

**头文件，例如**

ji\_license.h

**相关工具**

ev\_encdec 字符串混编工具

ev\_license license相关工具。

在算法容器内生成参考码；生成 license；校验license等

git地址：[git@git.extremevision.com.cn:ExtremeVision/license.git](mailto:git@git.extremevision.com.cn:ExtremeVision/license.git)

bin目录下就是不同版本license库提供的头文件和静态库，同时提供相关工具（直接运行会打印使用帮助）。

## 1.1生成参考码

参考码基于算法容器运行环境生成，以确定每个实例的唯一性。

### 1.1.1生成规则

* 生成参考码源文$reference\_origin，包括硬盘系列号+容器ID+容器网卡地址；
* 生成参考码$reference，例如hmac\_sha256 ($reference\_origin, $ key)；

**备注**

$ key可以是公钥或一个随机字符串。

上述规则仅供参考，实际可能有所不同，版本不同也会有所不同。

### 1.1.2函数说明

**int ji\_generate\_reference(char \*info, char \*reference, int \*version);**

/\*

函 数 名 : ji\_generate\_reference

功能描述 : 在既定运行环境生成参考信息,包括参考码和版本

函数参数 :

info: [可选参数] 输入输出参数,参考信息缓冲,json格式,

包括参考码和版本(建议分配(EV\_MESSAGE\_MAXLEN)

reference: [可选参数] 输入输出参数,参考码缓冲(建议分配EV\_ID\_MAXLEN)

version: [可选参数] 输入输出参数,参考码版本

返回参数 : 成功返回EV\_SUCCESS,其它表示失败

注 意: info一组,reference与version一组，两组至少传一组

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

当输入输出参数info不为空时，返回数据为json格式，例如

{

"version": 1,

"reference": "a67b4e7ce377c3adc42dbdb0cf6b2a82ac934553334faf07fe02c065c0db9342",

"disk\_serial": "94Q0A01CFTR6,94Q0A00TFTR6,94Q0A019FTR6,94Q0A00OFTR6",

"container\_id": "ef3ffa8711bb7403ddc0806bfd377fdb5ceee868f56c9cd45f309b8690b66520",

"mac": "02:42:ac:11:00:04"

}

## 1.2生成license

该函数有不同语言的实现版本。

### 1.2.1生成规则

* 用私钥对不仅限于参考码$reference进行签名，生成$license；
* 若增加时效性，添加时间戳信息，格式为”**签名时间:过期时间**”，例如

**1547088964:1549680964**

* 若增加qps,与增加时间戳实现一致；

### 1.2.2函数说明

**int ji\_generate\_license(const char \*privateKey, const char \*reference,**

**const char \*timestamp, const int \*qps,char \*license);**

/\*

函 数 名 : ji\_generate\_license

功能描述 : 根据摘要生成授权码，同时还可以指定有效期

函数参数 :

privateKey: [必选参数] 输入参数,私钥

reference: [必选参数] 输入参数,参考码

timestamp: [可选参数] 输入参数,指定有效期

qps: [可选参数] 输入参数,指定请求量上限

license: [必选参数] 输入输出参数,授权码缓冲(建议分配EV\_ID\_MAXLEN)

返回参数 : 成功返回EV\_SUCCESS,其它表示失败

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

## 1.3校验license

### 1.3.1校验规则

* 根据版本获取参考码$reference；
* 用公钥对$license确认签名，以判断license正确性;
* 加入license有效期及qps实现原理类似；

### 1.3.2函数说明

**int ji\_check\_license(const char \*pubKey, const char \*license, const char \*timestamp, const int \*qps,int version);**

/\*

函 数 名 : ji\_check\_license

功能描述 : 校验授权码

函数参数 :

pubKey: [必选参数] 输入参数,公钥

license: [必选参数] 输入参数,授权码

timestamp: [可选参数] 输入参数,指定有效期

qps: [可选参数] 输入参数,指定请求量上限

version: [必选参数] 输入参数,参考码版本

返回参数 : 成功返回EV\_SUCCESS,其它表示失败

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

## 1.4检验授权状态

### 1.4.1检验规则

在授权码校验成功时才真正检验

* 没有授权码时,返回EV\_FAIL;
* 有timestamp时,超过有效期,返回EV\_EXPIRE;
* 有qps时,超过请求量上限,返回EV\_OVERMAXQPS;
* 其它返回EV\_SUCCESS.

### 1.4.2函数说明

**int ji\_check\_expire();**

/\*

函 数 名 : ji\_check\_expire

功能描述 : 检验授权,有效期及请求量上限

函数参数 : 无参数

返回参数 : 已授权且在有效期内且未超过请求量上限时返回EV\_SUCCESS,其它表示失败或过期或超过请求量上限

备注 ：在授权码校验成功才真正检验

1.没有授权码时,返回EV\_FAIL;

2.有timestamp时,超过有效期,返回EV\_EXPIRE;

3.有qps时,超过请求量上限,返回EV\_OVERMAXQPS;

4.其它返回EV\_SUCCESS.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

# ev\_sdk同步更改

请参考ji.h相关文档

## 2.1ji\_init

**int ji\_init(int argc, char\*\* argv);**

例如

argc：4

argv[0]：$license

argv[1]：$timestamp

argv[2]：$qps

argv[3]：$version

**若没有时间戳，$timestamp为NULL或空字符；**

**若没有请求量限制，$qps为NULL或小于等于0；**

**建议授权失败返回 -999。**

## 2.2其它函数

ji\_create\_predictor 建议无授权时，返回NULL

ji\_destory\_predictor 暂时不用处理

ji\_calc\_frame

ji\_calc\_buffer

ji\_calc\_file

ji\_calc\_video\_file

建议以上四个函数，在第一步时调用ji\_check\_expire()函数，用于实时判断授权的状态。