公有云识别结果加密

1.识别结果加密说明

为保证用户的个人敏感信息，如身份证图片、以及识别出来的身份证图片文字信息不被泄露，对识别结果result中的数据进行加密传输，返回给用户。

2.加密方式

本次加密使用的是国密算法SM4,SM4分组密码算法是我国自主设计的分组对称密码算法，用于实现数据的加密/解密运算，以保证数据和信息的机密性。要保证一个对称密码算法的安全性的基本条件是其具备足够的密钥长度，SM4算法与AES算法具有相同的密钥长度分组长度128比特，因此在安全性上高于3DES算法。

此次加密采用的SM4的CBC模式，补码方式为PKCS5，密钥长度为128bit，使用的返回参数中request_id的后十六位，偏移向量iv为一个16字节的数组数据全为0。

3.调用说明

调用时，识别请求需补充识别结果加密请求参数result_encrypt=1

注：识别结果加密目前只适用于通用ocr所包含的接口，不包含查验相关的接口，且只有当识别正常返回（即error_code=0）时进行识别结果加密返回。

以下内容为营业执照接口的加密返回示例：

```
{
    "result":
"fjAfOvgsSqENUjo+mjm/kVN5RdbipsgaVGEe7fgJD9zN8KBpDe3QovvQjKwk+VVj8Y1WoI7gcFBysom
gBDvDFKui66IPajAOVM4UD2GOM2M+g3ORoorZLmtvVwbYW++dBgd6oWmHmVYa+ttzh2PHOiFe3tkj8Y3
h/tkC4SMzJvDzyGsvjmWZWehH/jdkotBlsPzaHG2cYMkIj9OmTlBJUa2YTXc3kqiTlQF5wX5rDDkKmMp
yRpjchWlS4juoAQMsktd3jEeEhVS8+kcPOJln5XazkAzp8n1pZdczM1cnasOerSjyCAmO6+ZI2MojQq9
H4xXwhvzs97LenelOCffPOsWv98iNK8c/TkWiF7L7DungLDSwVl7QsDEjui3dJ9lwTaer9EUdFmd9OBP
52O13bn2OBBdV+MaR8Te2hOvjFLGW9a3IWH34la2V25YaugDVBV2CFYeNiWixVjHwY75EjEeej8WL2L+
wbkkVXWSnScsiJTQSeW4EwxbR9DgF8sE93dYr8n4fNLrl+tQcrviwuvOTW7Y/QwynYvg+lXy+uZoX+OD
4WcGbH5CGluBYmMi/66Xtn76JYqm8cpiHauvVGBKArjYqYQyirRwOh9OnZBrYk6fdTqRy1UpGAFVZ9bl
OjV5+QnOe8hVpSm6Koj+vpaRQNk3TLxATi/e3K/j/LAv9cKc1aWlLj2WSA5UK3uKXiLGi1XSLIH7mXQl
hyBgNolwvpoQNyhD6aY/PY+XJYs6EvGBIsKZEue+lfLGW49AppG9OEVWFvtpJE3UbJVrLYrhxeFZFN76
WJVBra28+5RZIuyNcTF+e7dzx/fsE1nAenFBHn5qZN/UxCQqK7GlT3KTj4HlZZf+l/opge6QGqvwtvm1
43OqMHVmPNxTpjk/ck9SUb3OVXMn+3si39cUgqQQEXHrVcn0sVb1itRDd3/qqROExh1JlOYRB8sfA/B1
HVBcIF9nAtGz6WS2+XhP/9L+p/Y9uODxkKpdipiKydpGkznW6nu3yaqsaunhE6Iuypg+E31sAPaT3zcf
H2tp1NOwc1xZBufKsBjJABtQKCOgzO9uffSgx7y2EFOptzS6II/6SfEk5ttpX5mo9ysIcWMa8IpoO9BU
zOjFQypr7zOoOnt7OwXSghZXLlFzOQ8s3x69BkoWJzzuhGRZqOJyCmol4Imm2F+D/7XPiy7kSawuEyxe
ipS+Y4ipwmiQfOt49BO24AViROjqXIkO6+rtnQ/OPFZ9lAJA6fy7sNOPT7XoofQwj9lfnyKBx9NIg6jC
6DSbtWLDSM6M69dSHZjikcAbB2qbjRxpjd+Mn/m4phOrQ+vVf/gHkLgDpo5zmR7dU7Fy3vcd8RueFGha
8R4pfVcpzuE+2tkUbqNl3gsbPzjtCxzZTMpqquI/LbADUsEvnL+qzr5gLnSibaL1Y3dhFIdSoMt3nsVq
SJH8CFeGb+g8vdCal5gFc/jYqOMIsm+uqgYNX22Q+tD5xsKNnqEKzIZDcKfddiQJHZx+3Ov5coBL7V7d
KeErpkK3HpdqexHeGyyiBbWUdcQ9/DGOIUrRPsZou+cjZpoMcSunpePe8u31276EHQrQJpryKb7Rqkwa
uDSBlhWXb+d3IGF2gKshvmwrP8zp9jL7N+XAe3OOJLmSpLDqtR2ZYt2RjHLY5u2ymhZOt3Ag8qW0lhhB
u99cZDQF32dH6OHk3TlETi5jL6LVffPFchzNf7pLttbnWIWvOb7+T36OW9r6o6IVA/yK/qn886HqauAC
UFA+AOWGSZZIHhFXdu3T3HC/NW3t7SPLtsDkkm2CDA+K0iCdBCBi76Rkp+DLw7YByP3S1RfoGygu9lDE
93Y8T5VOFcbiw5etoS6GNecOmRoeqWnwitioh8LiNl1fD2uIlOm7CoXTXhTukeZAAaeMsweluJLzRM4D
YOdh7vjOxeQOO899OiUAK/r2yNmPquwD8EVScocvOW8ClnYb3MWaoKPlPv35rwRm+SSCeYPDnhJ8YG1L
To8T/Cu4tyc2+lvfJUZTDMIkyOSWp5i4/Qk2COGdzAzVttIbN5TdDHeDnUiq3KkQSUJDs92B1kIhgFpL
vaLM4IkldnPR9XsKLaBQtOa73N2i9ynO6PcE+68AAMaFi+R9BGjoIWbfs2XEbHOwhtdOY+qt2sxPkRVA
vztz2httonkpO28Jsq12OPxPbZY/cvmJr7uhMutKq/Du5sh6+Wqy68G4RPgMOl7zJvlgbiRoYWRk92AD
Zwznt/ASbun4auwhec7KFJsahOviiPQVIev9lhopmkCe5AtEVrQtZiYAMPIvJI578EpjVRixeJkFylRG
po2dUCGAPguWy86DW8T5aPXk4yIQx/93cjh2WdJE7QtWifBTg6XpcNPqlKOOjZztb+Jm3xrvVfW/VfhP
81JJqwQP16I4UhIFpVeC76RKOspA9CTfm2uXABBY5Qi9WdlBnYr2LgFh1dRd5DZyKPjFm6/zZzC9Piix
8XzPwA1bH6821jCU9UG1RklLiewOLNIeRBIc5A8pyE+K/GoIW72xnmsv/8XBliS2CZp7jC+PIOJoktLb
jhMBA7cX3abLGlEJVoMObwFEuWAzggJSmmEmyX2qCyT8d4Z58Oi2AqoVzB2tOwz4h+CfFmzIPGX1RjqqeF7sflTvsmLxeNPk4W2X1iDAJs71EhavFecvKom0xHnUM2DrfTQMiZ95w+gwwctRv6bzRCbea08ZidD4kzQoxgjBJpKXqaYQcOqj2BLtEOPXhDayYnjtwpX0YzzNtvy6+Tlf1DmdGoJFzmLxhv9npw74WQ989haRc6CY2oUFbkTUGjDifsbUQrJYcGalGKfv5l4FTsRFb8GNh6FyDL2mqEVX+PiGKdex8qz/0iZ2a42o9Gm7N4NMUDvlU5fOAQOuLf2savkk0XHhFguGj8nF71MPoG3AlN9G13JNnoqbrNfmHJ4K8mAlGdBV87wkkY5MeJ1+N+P7eXJnbtmGfJjTBAV+G1RTpRsZOKZQGhorGjwnr8Do7E1rI2mZSiEPzGRmbhxC6NBBmF+pE1rb8hf9tc15T+9VjyRjX7yY5W6uIcd1M/l4OQxmAGsMpQ8PRekBLHb6QLPoDeiNmbJD01XsJUdt5wCp5munTgyUOHELVfx21LPvQy91e+n+SOgrcqM1Emo3+/fjvlNG8Oom7tcv+F99HSM4pdTfIswpM+W1vOoaJZvaeGNNaP5Fe+XeBxjThAsG5+nHWTOniWrfYadD8lfVrLj6nwiqoUEzqvAwNZ8qEt/9Se7ZqWADIrz1iPTnm6LTmHXPJ8NG/Pgx8RqXgyxtwhDEFs88Y2wNmOqNp7HhxtYcv8rtxEeDJlHVdci5+Ki9LTqYrW3a8rOH
PekewwohtGnvEoe2mgwNiYrLbGuWOliDsDvw68zAgnAOCbtkrbrV2ROslXaN1n4BOci4Ltq873UXWBUG
rVyYUNSIUBNOeH1XwRdoOV2dtlXYpGg51nCaqSv8k3UOuWnIxUKR66x075zKMG4ZHXj6KT4ZTaXTo1F6
f3yv6R/w3feWODVShU3QRosVaBOdqhIOoRxY1GUkNwaP9e03JYIT5RrfgxgF8dfpgpCvKrg9UDkBFsxh
RCWsnplwfNwAnQP+qVvIPvV8DZwDOArZ9l+U5gbpaE46AHSaUagRMxsEnWd+uIOVK2Omfc4KhOek/fiL
AdgIwq59jwzITeY/AR87xR/EKMakclGGseDOet5G5e6tJEKcYfwspa3/SjReoB+XM0HxkwGdqfqC9Usw
ttD37QuISitJEyMI8zglWcHWnhPprs9AYhwdoZUmV4rOg4+Q5elIMPBEW8uYAzE/A7OtwI1sTU64Dvak
X6Y/uvV2Wbj/VAu64P4L5yntqqN5IwgimoIuWVMDIvJzdVuhLkfr1QjOcKs5BnPerO1+kUOivAnBWwhl
GrKxuUGjQgjWaJisz9I8xXmaiW5I=",
    "error_code": 0,
    "description": "识别成功",
    "request_id": "8F4F349583AB48C8AB122CC13ED7D912",
```

```
    "recognize_time": 464,
    "available_count": 5785,
    "version": null,
    "rotation": "0"
}
```

4.请求参数加密说明

补充请求参数image_sm4,作为加密后的图片请求参数，加密方式与第二节所示一致，其中密钥采用的是请求参数app_key中的前十六位作为固定参数，其余不变

5.加解密代码示例

引入依赖

```
<dependency>
    <groupId>cn.hutool</groupId>
    <artifactId>hutool-crypto</artifactId>
    <version>5.7.1</version>
</dependency>

import cn.hutool.core.util.CharsetUtil;
import cn.hutool.crypto.Mode;
import cn.hutool.crypto.Padding;
import cn.hutool.crypto.SmUtil;
import cn.hutool.crypto.symmetric.SM4;
import lombok.extern.slf4j.Slf4j;

import java.util.Arrays;


// SM4对称解密 CBC
    public static String sm4CBCDecrypt(String key, String cipherText) {
        long l = System.currentTimeMillis();

        //iv 为一个 16 字节的数组，数据全为0
        byte[] iv = new byte[16];
        Arrays.fill(iv, (byte) 0x00);
        SM4 sm4 = new SM4(Mode.CBC, Padding.PKCS5Padding, key.getBytes(), iv);
        String s = sm4.decryptStr(cipherText, CharsetUtil.CHARSET_UTF_8);
        log.info("sm4解密耗时:" + (System.currentTimeMillis() - l));
        return s;
    }

// SM4对称加密 CBC
    public static String sm4CBCEncrypt(String key, byte[] bytes)  {
        long l = System.currentTimeMillis();
        //iv 为一个 16 字节的数组，数据全为0x00
        byte[] iv = new byte[16];
        Arrays.fill(iv, (byte) 0x00);
        SM4 sm4 = new SM4(Mode.CBC, Padding.PKCS5Padding, key.getBytes(), iv);
        String s = sm4.encryptBase64(bytes);
        log.info("sm4加密耗时:" + (System.currentTimeMillis() - l));
        return s;
```

```
    }
```