开门见山地说，希望各位前端不要搜到这篇文章，因为我总感觉在小程序端去做RSA的加解密总有种脱了裤子放屁多此一举的感觉。但是需求来的时候总是说不准的，接下来我就给大家推荐一种在小程序端比较方便，操作不太多的RSA加解密方法给大家。

 我推荐的方法是使用：wxmp-rsa，Github上有他的正版介绍，他可以支持网页端和小程序，下面我主要介绍其在小程序上的使用方法。

**Wxmp-rsa的公钥使用的是PKC#8填充格式**

他是一个前端rsa加解密工具。

基于jsencrypt修改扩展功能。兼容小程序环境，压缩后60kb左右的大小，节省小程序空间。

支持超长文本加解密。

支持中文字符的加解密。

  正式开始前先确保自己的小程序编译环境可以使用npm命令

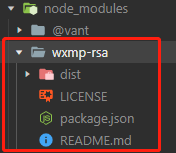


  1.安装

  选择小程序的调试器——》选择终端——》输入npm i wxmp-rsa -S进行安装



  安装完成后小程序目录里面会生成node\_modules文件夹，里面会有wxmp-rsa文件夹

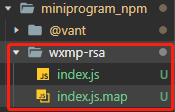


  2.构建npm

  安装完成后在编辑器左上角的菜单栏中找到工具——》构建npm



  完成后小程序文件目录会构建出miniprogram\_npm文件夹，里面会有wxmp-rsa文件夹



  3.具体使用方法

  在具体要实现的js文件中导入包，并完成实例化

// 导入包

import WxmpRsa from 'wxmp-rsa'

// 实例化rsa

const rsa = new WxmpRsa()

1

2

3

4

  4.加密

  加密需要使用公钥

// 定义待加密的字符串

const str = '{"id":"3254325423dsadsa"}'

// 定义公钥

const publicKey = `

-----BEGIN PUBLIC KEY-----

MIGeMA0GCSqGSIb3DQEBAQUAA4GMADCBiAKBgFnWSUwsmGawhMJ30z6y5li2jcf1

m7rPMZcwZOS3To8bk3OBaMGhVEc1F8GtJBbc1rn/HCLNL9zrCy21EefJON8tRFcY

HnpseZSzh+349lIhS+MFw9x4JUddwSPDyxwha929cKzMuVoftu3CJ+kqDBVvxLk7

iDBzUMqW3Kgehk2TAgMBAAE=

-----END PUBLIC KEY-----

`

// 设置公钥

rsa.setPublicKey(publicKey)

// 加密

const cryptStr = rsa.encryptLong(str)

console.log('加密后的结果：', cryptStr)

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

  5.解密

  解密需要使用公钥对应的私钥

// 定义私钥

const privateKey = `

-----BEGIN RSA PRIVATE KEY-----

MIICWgIBAAKBgFnWSUwsmGawhMJ30z6y5li2jcf1m7rPMZcwZOS3To8bk3OBaMGh

VEc1F8GtJBbc1rn/HCLNL9zrCy21EefJON8tRFcYHnpseZSzh+349lIhS+MFw9x4

JUddwSPDyxwha929cKzMuVoftu3CJ+kqDBVvxLk7iDBzUMqW3Kgehk2TAgMBAAEC

gYBRChPeyk/EOrHX912xLpLKLguh+LY9g1B50ScChzUvtTGDPZaxLQYoogVHKhfn

I9nzuOS5pBzsDX9tAO0hCQzqfHgqRjn+vEgm1Ui+f0E3BVRnhobcJKZpZqlvCBR5

Gu2+zlrY4SeGq3AuQSr/A5FiB5k0RgsvNycDTjqyg7TXGQJBAJoZ8Yr0zakxT1I8

lVqsFbeNPtt8FNG2UgIlIs9RL7aXhw+Y3sWtk/kbaOXafSofu0NcQYx4Km3M3kiP

lcNfTJ8CQQCVPcaRpu+mprRgHS6s76Z668NaFsjX04CUUa0kCrey+Nf/SJJ3BkRH

M7GllZWuI/RSXs/F5N38p5bfkn7QZqaNAkBy3dHJZW8DpgjdYOFnhAxwFK39BwGx

zHhWtv26kWbCcTKwsp+jtB4vunm3k+RmiN6aeGM35L6jt+kdJ0JYLmo7AkBJpRZb

wZj5D8Jqu3vQ8uGgPr9DsYKinkgQ6M0bv/4uXwWXf+Rmv7zpteSv5UTbjfp+uzKk

YO/6QWj+InhZto3xAkAOA0i702dLHm5elLWvht7UEYIDEW1+rYGdbthmJBvT9sZh

VKL954Y9hDzBWepjYsBiJnmIkgeladPnU5025/G/

-----END RSA PRIVATE KEY-----

`

// 设置私钥

rsa.setPrivateKey(privateKey)

// 解密

const originalStr = rsa.decryptLong(cryptStr)

console.log('解密后的原始数据：', originalStr)``

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

  6.注意事项

  例子中的加解密都是安装rsa标准的1024bit进行处理的，如果有特殊需要使用的是512bit或者2048bit的情况，可以在加解密前设置如下代码进行更改。总之只要公钥和私钥是对应的就没啥问题

rsa.default\_key\_size = 512

1

  以上就是微信小程序前端使用RSA加解密的方法了，大家使用的时候要去生成自己使用的公钥密钥哦，希望这篇文章能帮到大家。

————————————————

版权声明：本文为CSDN博主「LeeBrook0」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/weixin\_38611617/article/details/122731818