فناپ‌پلاس

FANAPPLUS

امضای دیجیتال (ارسال پیام)

نسخه ۱,۱,۰,۰

تاریخ ۱۳۹۷/۰۵/۱۷

فناپ‌پلاس
FANAPPLUS

فهرست

امضای دیجیتال

تمامی سرویس های سیستم ارسال و دریافت پیام، می بایست توسط برنامه توسعه‌دهنده امضا گردد. امضای یک فراخوانی سرویس تایید می کند که فراخوانی حتما توسط توسعه دهنده صورت گرفته است. برای اطلاعات بیشتر در مورد امضای دیجیتال می توانید به آدرس های زیر مراجعه نمایید:

- http://searchsecurity.techtarget.com/definition/digital-signature
- https://www.docusign.com/how-it-works/electronic-signature/digital-signature/digital-signature-faq

برای اینکه بتوانید یک درخواست سرویس را امضا کنید، نیاز به کلید خصوصی دارید. برای راحت تر شدن تولید جفت کلید های خصوصی و عمومی یک ابزار توسط این شرکت ارائه شده که با استفاده از آن می توانید جفت کلید های عمومی و خصوصی را تولید نمایید.

تولید کلیدهای عمومی و خصوصی

برای تولید کلیدهای عمومی و خصوصی، مراحل زیر را اجرا کنید:

۱. فایل KeyGenerators.rar را از آدرس زیر دانلود نمایید.

https://github.com/appson/payment-public/blob/master/v1.0.0/RSA%20Generator/src/Appson.Payment.KeyGenerator/bin/KeyGenerators.rar

۲. محتویات فایل را در مسیر مناسبی از حالت فشرده سازی خارج نمایید.

۳. برنامه Command Prompt را باز نموده و به این مسیر بروید.

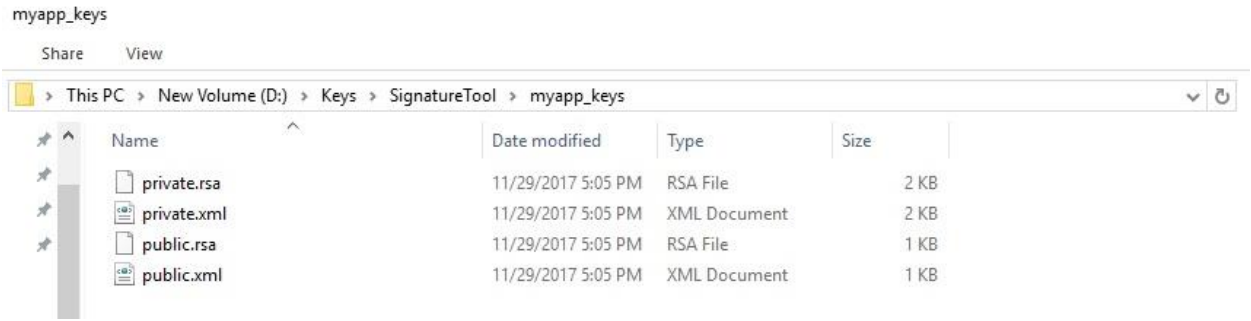
۴. عبارت Appson.Security.KeyGenerator.Console.exe را وارد کنید و کلید Enter را فشار دهید. از شما نام دایرکتوری کلید ها پرسیده می شود. تصویر 1 این فرایند را نشان می دهد.

```
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Windows\system32>d:

D:\>cd D:\Keys\SignatureTool

D:\Keys\SignatureTool>Appson.Security.KeyGenerator.Console.exe
Enter destination directory: myapp_keys
Keys generated successfully at the following address
D:\Keys\SignatureTool\myapp_keys
Continue (y/N)?
```

تصویر 1- تولید کلیدهای عمومی و خصوصی

۵. پس از وارد کردن نام دایرکتوری همانند تصویر 2، کلید خصوصی و عمومی ایجاد می شود.

برخی از کتابخانه ها از فرمت xml پشتیبانی می کنند و بعضی کتابخانه های دیگر از فرمت rsa پشتیبانی می کنند. در برنامه خود میتوانید از private.xml یا private.rsa استفاده کنید. اما در هنگام استفاده از توابع ارسال و دریافت پیام این شرکت می بایست محتویات public.xml را ارائه کنید.



تصویر 2- ایجاد کلید های عمومی و خصوصی در 2 فرمت متفاوت

امضای درخواست

هر پیام باید بر اساس ترکیب پارامترهای زیر، پیام فرمت شده‌ای را بسازد و آن را با کلید خصوصی خود امضا کند و در پارامتر قرار دهد.
ترکیب پارامترها از چپ به راست:

```
[Date],[Uid],[Sid],[ChannelType],[MessageType],[AccountId],[Content]
```

تاریخ درخواست باید UTC و با فرمت Iso 8601 باشد. (“yyyy-MM-ddTHH:mm:ss.fffZ”)

نمونه SignaturePattern :

```
var plaintext = $"{requestDateTime.ToStringIso8601Z()},
                  {requestUid},
                  {message.Sid},
                  {message.ChannelType},
                  {message.MessageType},
                  {message.AccountId ?? message.UserPhoneNumber},
                  {message.Content}";
```

```
var signatureParameter =      "2018-04-09T07:11:48.011Z,
                               9d6efd381534443e9e852abaf889d217,
                               914a0a94-a5b5-47c9-9a19-a8926643b1e9,
                               Imi,
                               Content,
                               53EF14CD98B64D2F814FA5E5428A,
                               This is a test message"
```

روش امضا کردن : الگوریتم مورد استفاده برای asymmetric cryptography، الگوریتم RSA است که الگوریتم hash آن نیز SHA1 است.

Signature = Sign(privateKey, signatureParameter);

Plaintext: 2018-05-28T14:00:28.363Z,50146e5507d74a8d935b30dda47c2678,1785317599f444449e550e7c93956de6,Imi,Content,09903024656

Signature:
Uh2g5Uf3+mAZGRccsmzm2qjhu27gbqLvbyuV3us3hsydhYi6tS/rIOG50yyDASugO4TwUGSyLpP2rqKD9hYCUnRazB8MjExcZJUP3grrrmeuES4nsNxEPG6Vb/2c3Tj5g8WqVZ0IO2N3Y1tY3HWprEwpujobbS5eTE08a6K0HUE9O0vU7MZ8NO2Cnjrj7M86qSw6vtig2SdZP2LpUJ4cG75jSZ/+X9v4r3koPn0q6+H2UY8/GNzBeoCURc4+h6RJXIzpU9hv/1HV2ng1/OB5o313IH/f/mJDy68FufJWCz2yTPzQ3tkKgqTOehmwv5DyWRqnxQm0YcK5iImue4FGfA==

jsonMessage: {"Uid":"50146e5507d74a8d935b30dda47c2678","Date":"2018-05-28T14:00:28.3631216Z","DateTime":"\/Date(-62135596800000-0000)\/","MessageType":"Unknown","ChannelType":"Unknown","Priority":"Unknown","Messages":[{"UserPhoneNumber":"09903024656","Content":"سلام","Sid":"1785317599f444449e550e7c93956de6","MessageType":"Content","ChannelType":"Imi","Priority":"Normal","Signature":"Uh2g5Uf3+mAZGRccsmzm2qjhu27gbqLvbyuV3us3hsydhYi6tS/rIOG50yyDASugO4TwUGSyLpP2rqKD9hYCUnRazB8MjExcZJUP3grrrmeuES4nsNxEPG6Vb/2c3Tj5g8WqVZ0IO2N3Y1tY3HWprEwpujobbS5eTE08a6K0HUE9O0vU7MZ8NO2Cnjrj7M86qSw6vtig2SdZP2LpUJ4cG75jSZ/+X9v4r3koPn0q6+H2UY8/GNzBeoCURc4+h6RJXIzpU9hv/1HV2ng1/OB5o313IH/f/mJDy68FufJWCz2yTPzQ3tkKgqTOehmwv5DyWRqnxQm0YcK5iImue4FGfA==}]}

Header: Content-Type application/json; charset=utf-8

- فرمت تاریخ ها بررسی شود.
- مسیج و آنچه ساین می شود دقیقا یکسان باشد.

سیگنچر با استفاده از متد زیر تولید شده است:

```
public static string Sign(string key, string text)
{
   using (var rsaProvider = new RSACryptoServiceProvider(CspParams))
   {
          rsaProvider.FromXmlString(key);
          var plainBytes = Encoding.UTF8.GetBytes(text);
          var encryptedBytes = rsaProvider.SignData(plainBytes, new
              SHA1CryptoServiceProvider());

          return Convert.ToBase64String(encryptedBytes);
   }
}
```

i. پارامتر اول کلید خصوصی می‌باشد.
ii. پارامتر دوم plaintext می باشد.
iii. پارامتر سوم plaintext است و با استفاده از کلید عمومی که در اختیار ما قرار گرفته چک می شود.

برای چک کردن سیگنچر از متد زیر استفاده شده است:

```
public static bool Check(string key, string signedText, string text)
{
   if (string.IsNullOrWhiteSpace(text)) return false;

   using (var rsaProvider = new RSACryptoServiceProvider(CspParams))
   {
          rsaProvider.FromXmlString(key);
          var encryptedBytes = Convert.FromBase64String(signedText);
          var plainInput = Encoding.UTF8.GetBytes(text);

          return rsaProvider.VerifyData(plainInput, new SHA1CryptoServiceProvider(),
                 encryptedBytes);
   }
}
```