

Forum:[Hack Forums](#)

Thread: [Generate asymmetric keys using custom seed?](#)

severos - January 29, 2017, 6:37 am

I'd like a way to generate the same asymmetric key from a custom seed, I know you can generate, export, import, etc. those keys, but it's not what I'm looking for.

So I want the user to be able to generate asymmetric key using an arbitrary string and be able to regenerate it over and over using the same string.

How is that possible using C#?

GuRuHaX - January 29, 2017, 1:35 pm

I think following is what you want:

PHP Code

```
<?php
public static Tuple<string, string> CreateKeyPair()
{
    CspParameters cspParams = new CspParameters { ProviderType = 1 };

    RSACryptoServiceProvider rsaProvider = new RSACryptoServiceProvider(1024, cspParams);

    string publicKey = Convert.ToBase64String(rsaProvider.ExportCspBlob(false));
    string privateKey = Convert.ToBase64String(rsaProvider.ExportCspBlob(true));

    return new Tuple<string, string>(privateKey, publicKey);
}
```

You can then use your public key to encrypt a message like so:

PHP Code

```
<?php
public static byte[] Encrypt(string publicKey, string data)
{
    CspParameters cspParams = new CspParameters { ProviderType = 1 };
    RSACryptoServiceProvider rsaProvider = new RSACryptoServiceProvider(cspParams);

    rsaProvider.ImportCspBlob(Convert.FromBase64String(publicKey));

    byte[] plainBytes = Encoding.UTF8.GetBytes(data);
    byte[] encryptedBytes = rsaProvider.Encrypt(plainBytes, false);

    return encryptedBytes;
}
```

And use your private key to decrypt like this:

PHP Code

```
<?php
public static string Decrypt(string privateKey, byte[] encryptedBytes)
{
}
```

```

CspParameters cspParams = new CspParameters { ProviderType = 1 };
RSACryptoServiceProvider rsaProvider = new RSACryptoServiceProvider(cspParams);

rsaProvider.ImportCspBlob(Convert.FromBase64String(privateKey));

byte[] plainBytes = rsaProvider.Decrypt(encryptedBytes, false);

string plainText = Encoding.UTF8.GetString(plainBytes, 0, plainBytes.Length);

return plainText;
}

```

Source :

<http://stackoverflow.com/questions/18850030/aes-256-encryption-public-and-private-key-how-can-i-generate-and-use-it-net>

Hope it helps !

?

severos - January 29, 2017, 8:06 pm

Thanks for your help, but that's not what I'm looking for, in your example if I run the program on my system, then close and run again it'll produce different keys, same thing if I run it on a different system, there's no 'seed' in there to generate the same keys over and over.

GuRuHaX - January 30, 2017, 1:30 am

(Jan 29, 2017 09:06 PM)severos Wrote: [\[CITE\]](#) Thanks for your help, but that's not what I'm looking for, in your example if I run the program on my system, then close and run again it'll produce different keys, same thing if I run it on a different system, there's no 'seed' in there to generate the same keys over and over.

Ok now i see what you want, but that wouldn't be safe at all to use a static one. It should always be dynamic (at least the private key). Its ok to let it be generated on the users side but even than it should be dynamic imo.

But to answer you should use lets say AES, encrypt a static value like 'SeverosApplication' and a salt value '123NowICanPee', and from that you could do the rest.

?

GuRuHaX - February 9, 2017, 9:48 am

Have you found a solution for what you wanted ?

?

Washi - February 9, 2017, 11:47 am

Well to be honest wouldn't that like defeat the purpose of the whole asymmetric key encryption in the first place? I mean the whole idea of asymmetric key encryption is that you have different keys for encrypting/signing and decrypting/authenticating. Because you are using the seed to generate the key-pair to both encrypt and decrypt, the seed becomes the one and only new key in this cryptographic algorithm as a whole, which basically negates the power of the asymmetric encryption, no?

Maybe I am misunderstanding you, and of course I don't know the context, but I don't see how this

is useful.

severos - February 9, 2017, 12:07 pm

(Feb 9, 2017 10:48 AM)GuRuHaX Wrote: [\[CITE\]](#) Have you found a solution for what you wanted ?

?The only solutions seems to generate the keys my self, thought of using the Rfc random byte generator class to generate the parameters and validate if they can be used as RSA keys.

(Feb 9, 2017 12:47 PM)Washi Wrote: [\[CITE\]](#) Well to be honest wouldn't that like defeat the purpose of the whole asymmetric key encryption in the first place? I mean the whole idea of asymmetric key encryption is that you have different keys for encrypting/signing and decrypting/authenticating. Because you are using the seed to generate the key-pair to both encrypt and decrypt, the seed becomes the one and only new key in this cryptographic algorithm as a whole, which basically negates the power of the asymmetric encryption, no?

Maybe I am misunderstanding you, and of course I don't know the context, but I don't see how this is useful. I understand your point, however I'm dealing with users who have absolutely no idea about encryption/programming and they're not good with computers at first place. I'm 100% sure if I generate the key for them and save it on a file they'll manage to loose it eventually. The product I'm working on requires asymmetric encryption as part of licensing system, so loosing the private key will prevent them from creating new license without major changes that will screw older users. So using a password as a seed to generate the key seems like the logical solution since they're more likely to remember a password than keep a file safe.

morphine1337 - February 9, 2017, 4:49 pm

Can't you just put them on a cloud somewhere ?

severos - February 9, 2017, 8:23 pm

(Feb 9, 2017 05:49 PM)morphine1337 Wrote: [\[CITE\]](#) Can't you just put them on a cloud somewhere ?

I would have done it if it's my keys, but I shouldn't be able to touch them at any point otherwise I'd be able to generate licenses my self which could screw the contract

Ninja - February 10, 2017, 10:05 am

(Feb 9, 2017 01:07 PM)severos Wrote: [\[CITE\]](#) The only solutions seems to generate the keys my self, thought of using the Rfc random byte generator class to generate the parameters and validate if they can be used as RSA keys.

I understand your point, however I'm dealing with users who have absolutely no idea about encryption/programming and they're not good with computers at first place. I'm 100% sure if I generate the key for them and save it on a file they'll manage to loose it eventually. The product I'm working on requires asymmetric encryption as part of licensing system, so loosing the private key will prevent them from creating new license without major changes that will screw older users. So using a password as a seed to generate the key seems like the logical solution since they're more likely to remember a password than keep a file safe.

Its almost never the users/buyers of the program that crack it, they post it on cracking forums for other people to crack, so it is a bad idea probably.

severos - February 10, 2017, 10:15 am

(Feb 10, 2017 11:05 AM)Ninja Wrote: [\[CITE\]](#) Its almost never the users/buyers of the program that crack it, they post it on cracking forums for other people to crack, so it is a bad idea probably.

It'll be cracked eventually if it becomes big, I'm just making it a bit harder, so instead of simply trying a password to decrypt AES encrypted file, you'll have to generate RSA key and check if it works, so should slow down the cracking a bit.

I agree it's a bad idea to use it this way, probably will just zip and embed the file to prevent simple

copy-paste, can't seem to find an easy licensing system that fits the budget for project.