About transfer of digital signature, cipher related files (2)

On iPhone/iOS, it seems that some special measures have been taken to transfer files and accept transfers for files whose file extensions are as follows:

cer

crt

der

pem

pfx

Therefore, it is impossible to transfer or accept files with such extensions.

rsac does not address this problem at all positively.

But it uses files with these extensions regularly.

Therefore, you may feel anxious about not dealing with anything.

Files with these extensions are very common in digital signatures and cipher related.

Usually, no special measures are taken and no special measures need to be taken.

The following is a brief description of what these extensions are like.

cer, crt

It is an X. 509 public key certificate.

An X.509 public key certificate is simply a digital signature that someone's public key is subject to signing.

In general, X.509 certificates are not confidential.

Generally, X.509 certificates do not contain anything that needs to be kept secret.

The meaning of the extension "cer" is the head of "Certificate".

The encoding method of ASN.1 syntax includes a method called cer,
but it is not derived from it.

The dangerous thing about X. 509 public key certificates is the only thing that

> "put a self-signed certificate that you do not intend to use as a root certificate in the root certificate repository.”

Other than this, X. 509 certificates are usually not at risk.

Digital signatures, including X. 509 certificates, are meant to be visible to people.

There is no such thing as concealment in digital signatures.

Also, PKCS#12 is sometimes referred to as a "certificate," but PKCS#12 is absolutely impossible to disclose to others.

This is because PKCS#12 contains the private key of public key cipher.

To call PKCS#12 a "certificate" is exactly misuse.

There is no such thing that a well-written document calls PKCS#12 a "certificate".

It is only a strange document that calls PKCS#12 "certificate".

der

This implies ASN.1 syntax.

For encoding ASN.1 syntax, usually cer, der and per are used.

The file extension of der is often used when filing something that the der method is used to encode ASN.1 syntax.

The extension alone can not tell at all what the contents of the file with the extension der are.

It do not know at all whether it should be kept confidential or whether it contains something that needs to be kept secret.

The above extension .cer, which means X.509 public key certificate,
and cer of the encoding method of ASN.1 syntax are different things at all.

pem

A file that has been textified in Privacy Enhanced Mail format.

The pem file has the same appearance as e-mail.

Handling is equivalent to text file.

The extension alone can not tell if it should be kept secret or if it contains something that needs to be kept secret.

In general, this method is used to convert encryption keys, ciphertexts, digital signatures, etc. into text files.

If something that needs to be kept secret is stored in the pem file,
it is usually encrypted before it is converted to Privacy Enhanced Mail format.

An unencrypted private key is sometimes converted into a Privacy Enhanced Mail format, but it is quite rare.

pfx

It is about PKCS#12.

The extension .p12 is often used.

PKCS#12 is usually never revealed to others.

PKCS#12 usually contains the private key and public key of public key cipher.

The private key of public key cipher is something that can never be revealed to others.

Handling requires great care.

PKCS#12 has not been updated for a long time.

Therefore, the PKCS#12 default encryption method is obsolete.

The contents of PKCS#12 are usually encrypted.

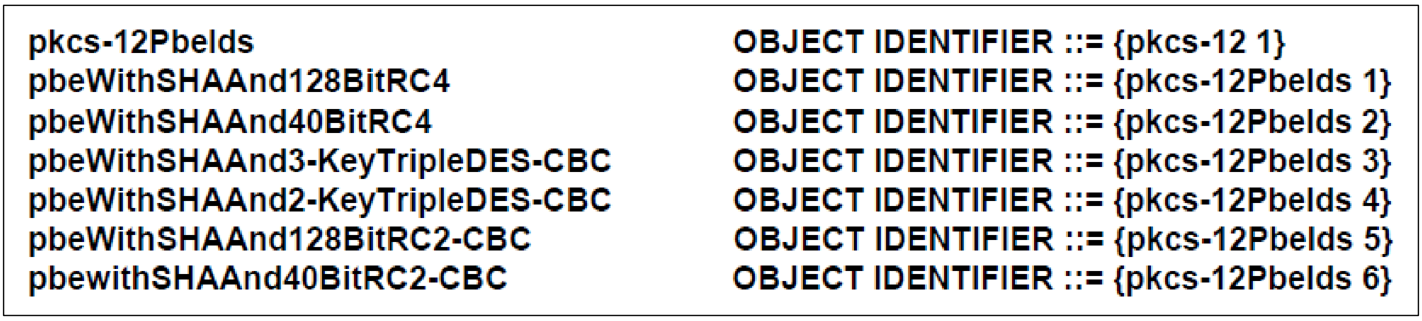
However, PKCS # 12's default encryption method is now the only one that is said to be able to be broken in a short time.

So now, the default PKCS#12 is, in effect, just like no encryption.

It is such a thing that the private key of the public key cipher, which can not be revealed to anyone else, is virtually unencrypted.

Please be aware that the default PKCS#12 is very dangerous because of this.

If this point seems to be dangerous, it is safe to encrypt PKCS#12 itself with AES etc.



```
pkcs-12PbeIds                        OBJECT IDENTIFIER ::= {pkcs-12 1}
pbeWithSHAAnd128BitRC4               OBJECT IDENTIFIER ::= {pkcs-12PbeIds 1}
pbeWithSHAAnd40BitRC4                OBJECT IDENTIFIER ::= {pkcs-12PbeIds 2}
pbeWithSHAAnd3-KeyTripleDES-CBC      OBJECT IDENTIFIER ::= {pkcs-12PbeIds 3}
pbeWithSHAAnd2-KeyTripleDES-CBC      OBJECT IDENTIFIER ::= {pkcs-12PbeIds 4}
pbeWithSHAAnd128BitRC2-CBC           OBJECT IDENTIFIER ::= {pkcs-12PbeIds 5}
pbewithSHAAnd40BitRC2-CBC            OBJECT IDENTIFIER ::= {pkcs-12PbeIds 6}
```

Although the Copy of PKCS#12 specification, PKCS#12's default encryption method is the only these six.

All of these are said to be able to be broken in a short time.

The default PKCS#12 is, in effect, just like no encryption.

Please be aware a such thing.

Also, PKCS#12 itself is a general-purpose data storage with authentication function.

So to speak, PKCS#7 with authentication function is PKCS#12.

PKCS#12 itself is not a dedicated container for private keys and public keys of the public key cipher.

It is only often (most) used this way.

But it is not exclusive.