

# Re: Logon with Digital Siganture (PKI/OCES – or what else they're called)

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2005-09/msg03007.html>

---

- *From:* [lynn@xxxxxxxxxx](mailto:lynn@xxxxxxxxxx)
  - *Date:* 24 Sep 2005 12:49:50 -0700
- 

Martin Høst Normark wrote:

- > Hi everyone
- >
- > Has anyone got the least experience in integrating the Digital Signature
- > with an ASP.NET[C#] Web Application?
- >
- > Here in Denmark, as I suppose in many other countries, they're promoting the
- > digital signature. A lot of people already has one, to do their taxes, and
- > much more. I have to use for a business-to-business e-commerce solution,
- > where it's vital that the right user is being logged on, and not give his
- > username and password to a colleague...
- >
- > Due to the Digital Signatures usage, companies are very aware of which
- > employees has access to tax, VAT and things like that – and I can make a
- > more secure web application...
- >
- > Anyone with just a good idea, own experiences, good links, or something?

One of the issues has been confusing identification and authentication.

the basic technology is asymmetric key cryptography; what one key (of a key-pair) encodes, the other key decodes (to differentiate from symmetric key where the same key is used for encryption and decryption).

there is business process defined called public key ... where one of the key (of an asymmetric key pair) is defined as public and made freely available and the other is identified as private and kept confidential and never divulged.

there is a business process defined called digital signature ... where the originator calculates the hash of a message/document and encodes it with their private key resulting in something called a digital signature. they then transmit the message/document along with the digital signature. The recipient recalculates the hash on the received message/document, decodes the digital signature with the (corresponding) public key and compares the two hashes. If they are

## Re: Logon with Digital Signature (PKI/OCES – or what else they're called)

equal, then the recipient can assume:

- 1) the message/document hasn't been modified since signing
- 2) "something you have" authentication, i.e. the originator had access to and use of the private key.

Basically, recipients build up a trusted repository of public keys used for verifying digital signatures.

Digital signatures have also been used to upgrade existing shared-secret

<http://www.garlic.com/~lynn/subpubkey.html#secret>

where the public key is registered in lieu of a pin/password and instead of matching pin/password, the public key is used to verify the originator's digital signature.

the most world-wide pervasive authentication infrastructure is possibly RADIUS (extensively used by ISPs and other organizations for integrated administrative authentication, authorization, and accounting). This was originally been password based infrastructure ... but has been upgraded with other technologies, like registering public keys in place of passwords ... and the method of authentication can be selected on an account-by-account basis.

<http://www.garlic.com/~lynn/subpubkey.html#radius>

another widely deployed (originally password) authentication infrastructure is KERBEROS ... originally developed as part of MIT's project athean (my wife and I periodically did project athena technology audits in the late 80s). You find KERBEROS integrated into lots of distributed infrastructures like windows and most unix flavors. Originally, the internet draft for upgrading KERBEROS to digital signature verification PK-INIT ... specified simply registering public keys in lieu of passwords.

Another business process related to digital signatures and public keys that evolved involves certification authorities (CAs), digital certificates, and PKIs. The paradigm is analogous to the "letters of credit" (or letters of introduction) from the sailing ship days. The design point is the offline email environment from the early 80s, where the recipient dials-up their local (electronic) post office, exchanges email, and hangs up. They then may be faced with first-time communication from complete stranger ... with no local information about the stranger and no resources available for obtaining information about the stranger.

The idea is that there are these things called trusted certification authorities that certify information about strangers and create digitally signed digital certificates that contains the certified information. Recipients (or relying parties) are expected to load the public keys of certification authorities into their local trusted

public key repositories.

Now when a stranger signs a message/document, they can transmit the message/document, their digital signature, and their digital certificate. The recipient will (hopefully) have the public key of the certification authority in their local trusted public key repository .... and can verify the CA's digital signature on the digital certificate. From this, the recipient can supposedly trust the information in the digital certificate. The recipient retrieves the stranger's public key (that has also been included in the digital certificate) and verifies the stranger's digital signature. The recipient now supposedly can use information about the stranger included in the digital certificate to determine what to do next.

One of the issues in the early 90s, was the x.509 identity certificate standard ... that also included something called the non-repudiation bit.

Basically certification authorities were looking at increasing the value of digital certificates that they were selling. First they were advocating that x.509 identity digital certificates be included on all digitally signed authentication events .... turning even the most simple authentication operation into a heavy duty identification operation (and not just simply limited to first time communication between strangers that had no other way of finding information about the other party ... either offline or online).

The other issue is that the certification authorities didn't necessarily know at the time they were creating an x.509 identity certificate ... exactly what information that all possible recipients might be interested in. As a result there was some direction to include enormous amount of personal information in these x.509 certificates (grossly aggravating the scenario of turning every trivial authentication operation into a heavy duty identification operation).

Then there was the non-repudiation flag. This possibly was the result of some semantic confusion where the term "digital signature" and the term "human signature", both contain the word "signature". The non-repudiation flag in a digital certificate supposedly met that the digital signature effectively carried the weight of a human signature .... i.e. human intent, read, understood, agrees, approves, and/or authorizes what was digitally signed.

It eventually became obvious that the setting of a flag in a digital certificate by a certification authority possibly months in the past ..... could not actual guarantee that a human has read, understood, agrees, approves, and/or authorizes something. As a result, the non-repudiation flag has become severely depreciated.

The other gap in the PKI protocol with respect to non-repudiation flag was that there is nothing in standard PKI protocols that can prove what specific digital certificate a person actually attached to any

## Re: Logon with Digital Signature (PKI/OCES – or what else they're called)

specific message. Given that a person might have two different digital certificates for the same public key ... one with the non-repudiation flag and one w/o the non-repudiation flag, then (in theory) the recipient need only be able to find and produce the digital certificate with the non-repudiation flag ... to demonstrate that what had been digitally signed is bound by human signature and non-repudiation rules. Again, after some real world experience, it quickly became evident that such a scenario was outlandish.

Going into the mid-90s, some number of institutions were starting to realize that x.509 identity certificates, grossly overloaded with enormous amounts of personal information represented significant privacy and liability issues. As a result, you started to see something called relying-party-only certificates

<http://www.garlic.com/~lynn/subpubkey.html#rpo>

the first ones that I'm aware of were by a large german financial institution.

The issue is that the institution places all the information about the individual in an accessible database and the digital certificate is purely loaded with the index pointer to the database entry and the individual's public key. It turns out that it became trivial to prove that relying-party-only digital certificates are redundant and superfluous; in part because it violates the fundamental design point originally used to justify PKI and digital certificates (the recipient had no other way of obtaining information about the originating entity). Given that the recipient has to use the database index to access the database entry, by definition they already have all the information that might be represented by a digital certificate.

In such situations, it is trivial to eliminate the redundant and superfluous digital certificates and return to the original digital signature authentication design ... using the information that the recipient already has access to.

<http://www.garlic.com/~lynn/subpubkey.html#certless>

note that sometime after the original pk-init draft for Kerberos, it was updated to add the possibility of supporting PKI-based operations (as opposed to simple, straight-forward registrations of public keys in lieu of password)

so that it would be possible for total strangers to logon to your system

(as per the original design point justifying PKI, certification authorities, and digital certificates).

As an aside ... there is further problem with trying to use digital signatures for both authentication as well as indication of human signatures involving intent, read, understood, agrees, approves, and/or authorizes. I've referred to this as a dual-use attack/vulnerability.

Many of the digital signature authentication infrastructures involve

Re: Logon with Digital Signature (PKI/OCES – or what else they're called)

Re: Logon with Digital Signature (PKI/OCES – or what else they're called)

servers transmitting random data to the client for digital signing (as a countermeasure to replay attacks). The client digitally signs the random data (w/o ever having read, understood, agrees, approves, and/or authorizes) and returns the digital signature. The issue is that an attacker might include some valid transaction or contract in lieu of random bits, the client then applies a digital signature to the non-so-random bits ... and the attacker then uses the information and digital signature as proof of a valid transaction/contract.

some number of past collected posts on electronic signature legislation (my wife and I were brought in to help write the cal. electronic signature and then the fed. electronic signature legislation), non-repudiation, and human intent ... as well as common for identification and authentication to be frequently confused  
<http://www.garlic.com/~lynn/subpubkey.html#signature>

here are posts that describes two-factor authentication involving a chipcard performing digital signature ("something you have" authentication) and a PIN ("something you know" authentication) ... where the entering of the PIN can also be used as an indication of "human signature". The scenario is that the chipcard calculating a digital signature has none of the characteristics required for establishing human intent, read, understood, agrees, approves, and/or authorizes. However, a certified terminal can display a message that say "enter you PIN if you agree to the transaction". The physical hitting of keys in response to a message can be used to establish human intent (in addition to knowing the correct PIN being used as a form of "two factor" authentication). The interesting aspect is that it is the entering of the PIN that is the basis for human signature and not the generation of a digital signature (where possibly because the term "digital signature" and "human signature" both contain the word "signature" that gives rise to frequent confusion).

<http://www.garlic.com/~lynn/aadsm21.htm#3> Is there any future for smartcards?

<http://www.garlic.com/~lynn/aadsm21.htm#5> Is there any future for smartcards?

<http://www.garlic.com/~lynn/aadsm21.htm#13> Contactless payments and the security challenges

some past posts on digital signature dual-use attack

<http://www.garlic.com/~lynn/2004i.html#17> New Method for Authenticated Public Key Exchange without Digital Certificates

<http://www.garlic.com/~lynn/2004i.html#21> New Method for Authenticated Public Key Exchange without Digital Certificates

<http://www.garlic.com/~lynn/aadsm17.htm#57> dual-use digital signature vulnerability

<http://www.garlic.com/~lynn/aadsm17.htm#59> dual-use digital signature vulnerability

<http://www.garlic.com/~lynn/aadsm18.htm#1> dual-use digital signature vulnerability

Re: Logon with Digital Siganture (PKI/OCES – or what else they're called)

<http://www.garlic.com/~lynn/aadsm18.htm#2> dual–use digital signature vulnerability  
<http://www.garlic.com/~lynn/aadsm18.htm#3> dual–use digital signature vulnerability  
<http://www.garlic.com/~lynn/aadsm18.htm#56> two–factor authentication problems  
<http://www.garlic.com/~lynn/aadsm19.htm#41> massive data theft at MasterCard processor  
<http://www.garlic.com/~lynn/aadsm19.htm#43> massive data theft at MasterCard processor  
<http://www.garlic.com/~lynn/aadsm20.htm#0> the limits of crypto and authentication  
<http://www.garlic.com/~lynn/aadsm21.htm#5> Is there any future for smartcards?  
<http://www.garlic.com/~lynn/aadsm21.htm#13> Contactless payments and the security challenges  
<http://www.garlic.com/~lynn/2005.html#14> Using smart cards for signing and authorization in applets  
<http://www.garlic.com/~lynn/2005b.html#56> [Lit.] Buffer overruns  
<http://www.garlic.com/~lynn/2005e.html#31> Public/Private key pair protection on Windows  
<http://www.garlic.com/~lynn/2005g.html#46> Maximum RAM and ROM for smartcards  
<http://www.garlic.com/~lynn/2005m.html#1> Creating certs for others (without their private keys)  
<http://www.garlic.com/~lynn/2005m.html#11> Question about authentication protocols  
<http://www.garlic.com/~lynn/2005o.html#3> The Chinese MD5 attack

---

• **References:**

- ◆ **[Logon with Digital Siganture \(PKI/OCES – or what else they're called\)](#)**  
    ◇ From: Martin Høst Normark

- Prev by Date: **[Re: Regex – Memory performance](#)**
- Next by Date: **[Saving Tiff as 16 bpp ?](#)**
- Previous by thread: **[Re: Logon with Digital Siganture \(PKI/OCES – or what else they're called\)](#)**
- Next by thread: **[Reflection, Custom Attributes, Inheritance Advice Needed](#)**
- Index(es):
  - ◆ **[Date](#)**
  - ◆ **[Thread](#)**