Digital Signatures and DSA

A digital signature is proof that someone has sent a message. (Goal: to prove that no one else could have penned the message, so that we can prove who wrote it, after receipt.)

Most simple technique is to use public key encryption with two sets of keys:

Alice (PR_A, PU_A) and Bob (PR_B, PU_B)

If Bob is sending to Alice and Bob wants to prove that Bob sent it, he can do this:

Step 1: $E(M, PU_A)$, this makes sure only Alice can read it.

Step 2: $E(E(M, PU_A), PR_B)$, the only person who can encrypt with Bob's private key, is Bob.

This is what Bob sends to Alice.

Then Alice decrypts using Bob's public (and in fact, anyone could do this step, since Bob's public key is public…)

$D(E(E(M, PU_A), PR_B), PU_B)$ … if nothing has been tampered with, this should yield:

$E(M, PU_A)$

Then Alice can use her private key to get the message, M.

This does everything we want. It allows only Alice to read the message and proves that Bob sent it.

Key downside: This is very slow…public key encryption of long messages is quite slow already, and this doubles the time spent.

What most digital signatures do instead: instead of encrypting the whole message, we hash the message and then encrypt that via a digital signature scheme. (The signature itself is pretty short and therefore will take less time to produce…)

For most digital signature schemes, we assume that we are using some hash function. The details of that function don't matter, only we must know the output bit size.

NIST Digital Signature Algorithm

This is more complicated than RSA or El Gamal. I don't have code for this…

Digital Signature is separate to the encryption (confidentiality). This is just a way to prove that a particular person really sent the message.

Global Public Key Elements

p = prime number, $2^{L-1} < p < 2^L$, (L bits exactly), $512 \leq L \leq 1024$ and L is a multiple of 64.

q = prime divisor of p-1, where $2^{N-1} < q < 2^N$, so q has N bits.

$g = h^{(p-1)/q} \bmod p$, where h is any integer $1 < h < p-1$ such that $h^{(p-1)/q} \bmod p > 1$.

User's Private Key

x, where $0 < x < q$ should be random (notice x's smaller range)

User's Public Key

$y = g^x \bmod p$

User's Per Message Secret Number

k, where $0 < k < q$ should be random and definitely not related to x!!!

To Sign a Message M

The digital signature is an ordered pair (r, s). Here is what is sent as r and and what is sent as s:

$r = (g^k \bmod p) \bmod q$ (r must be in between 0 and q-1)

$s = [\ k^{-1}(H(M) + xr)\ ] \bmod q$

To Verify a signature (r', s'), want to see if this is the real signature (r, s).

$w = (s')^{-1} \bmod q = [\ k^{-1}(H(M) + xr)\ ]^{-1} \bmod q = k(H(M) + xr)^{-1} \bmod q$

$u_1 = [H(M')w] \bmod q = H(M)\ (k(H(M) + xr)^{-1}\ ) \bmod q$

$u_2 = (r')w \bmod q = rk(H(M) + xr)^{-1} \bmod q$

$v = [(g^{u1}y^{u2}) \bmod p] \bmod q$

$= g^{H(M)(k(H(M) + xr)^(-1)}y^{rk\ (\ H(M) + xr)^(-1)}$

$= g^{H(M)(k(H(M) + xr)^(-1)}g^{rx\ (\ kH(M) + xr)^(-1)}$

$= g^{H(M)(k(H(M) + xr)^(-1)\ +\ rx\ (\ kH(M) + xr)^(-1)}$

$= g^{(k(H(M) + xr)^(-1)\ (H(M)+\ rx)}$

$= g^{(k(alpha))}$, where alpha is equivalent to 1 mod q

$= g^{(k(qw+1))}$, where w is just an integer.

= $g^{(k(qw+1))}$ first mod p, then mod q

= $(g^{kqw})(g^{k})$,

At this point, we must remember that g = $h^{\frac{p-1}{q}}$. Plugging in we get:

=$h^{(\frac{p-1}{q})kqw} g^{k}$

The q's cancel, so we have:

=$h^{(p-1)kw} g^{k}$

Due to Fermat's theorem, the first term is equivalent to 1 mod p, so we have

$\equiv g^{k} (mod\ p)$

as desired.

When we verify the signature, what we expect is that this calculation will give us

g^{k} mod p mod q.

If it doesn't then, someone changed either the message or something else.

Example

p = 137, q = 17 Calculate a digital signature (r, s) for this system when you choose k = 13, the public generator h = 2, so g = $2^{136/17} = 2^{8}$ = 119 mod 137, the private key x = 3, public key y = 119^{3} = 59 and H(M) = 12.

r = 119^{13} mod p mod q = 133 mod 137 mod 17 = 14

For s, we need 13^{-1} mod 17,

Extended Euclidean algorithm…(you can do this on your own), but when you finish, you find that

$k^{-1} = 4$ (mod 17)

s = [4(12 + 3*14)] mod 17 = 216 mod 17 = 12

To verify…

w = 12^{-1} mod 17 = 10 (since 12*10 = 120 = 1 mod 17)

u1 = H(M)w mod q = 12(10) = 120 = 1

u2 = rw mod q = 14*10 mod 17 = 140 mod 17 = 4

$g^{u1}y^{u2} = 119^{1}59^{4}$ = 119*12117361 = 1441965959 mod 137 mod 17 = 133 mod 17 = 14

So the value checks out!!!