Digital signatures

2010-10-19

RSA signatures

RSA private key: (n, d) and corresponding public key: (n, e) where $n = pq$, for primes p, q and $de \equiv 1 \pmod{\varphi(n)}$.
(Remember that $\varphi(n) = (p-1)(q-1) = pq + 1 - p - q$.)

Signing a message m: The signature is

$$y = m^d \bmod n.$$

Verification of the signature: Compute

$$m = y^e \bmod n$$

and check that m makes sense.

RSA hash-and-sign

For this common method, let h be a hash function and ensure that n is bigger than a hash value.

Signing a message m: The signature of the message m

$$y = h(m)^d \bmod n.$$

Verification of the signature: Check that

$$y^e \equiv h(m) \pmod{n}.$$

Advantages:

- You can sign longer messages
- Detached signature: The signature can be distributed separate from the message.

Blind RSA signature

Let Alice's private key be (n, d) and her public key be (n, e).

Bob wants Alice to sign a message $m \in \mathbb{Z}_n^*$ without her knowing what she signed.

He picks a random nonce $k \in \mathbb{Z}_n^*$ and gets Alice to sign $x = k^e m \bmod p$.

The resulting signature is $y \in \mathbb{Z}_n^*$,

$$y \equiv x^d \equiv (k^e m)^d \equiv k^{ed} m^d \equiv k m^d \bmod p,$$

and from this Bob can compute the signature

$$m^d \equiv k^{-1} y \pmod p.$$

RSA key use: Be careful!

- Never use the same RSA key for encryption/decryption and for signing.
- Never ever use a regular signing key for blind signing.
- In fact, create a separate key for each blind signing application.
- And finally, make the intended use of each of your RSA keys very clear. (See: PKI and certificates, etc.)

ElGamal signature scheme

Private ElGamal key: (p, α, a). Corresponding public key: (p, α, β).
Here p is a prime, α a generator of $\mathbb{Z}_p^*$, $a \in \mathbb{Z}_{p-1}$, and $\beta = \alpha^a \bmod p$.

The signature on a message $m \in \mathbb{Z}_{p-1}$ is a pair $(r, s) \in \mathbb{Z}_p^* \times \mathbb{Z}_{p-1}$
where $k \in \mathbb{Z}_{p-1}^*$ is a random nonce,

$$r \equiv \alpha^k \pmod{p}, \quad m \equiv sk + ar \pmod{p-1}.$$

Note that then

$$\alpha^m \equiv \alpha^{sk+ar} \equiv r^s \beta^r \pmod{p}.$$

To verify the signature, therefore, check that

$$\alpha^m \equiv r^s \beta^r \pmod{p}.$$

The Digital Signature Algorithm (DSA) (1)

DSA private key: (p, q, α, a). Corresponding public key: (p, q, α, β).
Here p and q are primes, $q \mid p-1$,

$$\alpha = g^{\frac{p-1}{q}} \bmod p, \quad \beta = \alpha^a \bmod p$$

where g is a generator of $\mathbb{Z}_p^*$ (which is not needed after computing α).

Note that $\alpha^q \equiv 1 \pmod{p}$, and $\alpha^k \not\equiv 1 \pmod{p}$ for $1 \leq k < q$.
In other words, the multiplicative period of α (modulo p) is q.

The DSA standard requires that q is a 160 bit prime. p will have at least 512 bits, usually many more.

The Digital Signature Algorithm (DSA) (2)

DSA private key: (p, q, α, a). Corresponding public key: (p, q, α, β). Here p and q are primes, $q \mid p-1$,

$$\alpha = g^{\frac{p-1}{q}} \bmod p, \quad \beta = \alpha^a \bmod p$$

and $\alpha \in \mathbb{Z}_p^*$ has period q.

The signature of a message $m \in \mathbb{Z}_q$ is $(r, s) \in \mathbb{Z}_q \times \mathbb{Z}_q$ where

$$r \equiv (\alpha^k \bmod p) \pmod q,$$
$$ks \equiv m + ar \pmod q$$

and $k \in \mathbb{Z}_q^*$ is a random nonce.

The Digital Signature Algorithm (DSA) (3)

The signature of a message $m \in \mathbb{Z}_q$ is $(r, s) \in \mathbb{Z}_q \times \mathbb{Z}_q^*$ where

$$r \equiv (\alpha^k \bmod p) \pmod q,$$
$$ks \equiv m + ar \pmod q$$

and $k \in \mathbb{Z}_q^*$ is a random nonce.

Note that then

$$\alpha^k \equiv \alpha^{s^{-1}m + s^{-1}ar} \equiv \alpha^{s^{-1}m} \beta^{s^{-1}r} \pmod p.$$

To verify the signature, check that

$$r \equiv \left(\alpha^{s^{-1}m} \beta^{s^{-1}r} \bmod p\right) \pmod q.$$