ElGamal signature scheme

From Wikipedia, the free encyclopedia.

The **ElGamal Signature scheme** is a digital signature scheme which is based on the difficulty of computing discrete logarithms. It was described by Taher ElGamal in 1984. The ElGamal signature algorithm described in this article is rarely used in practice. Much more popular is a variant known as the Digital Signature Algorithm. The ElGamal signature scheme must not be confused with the ElGamal encryption which was also invented by Taher ElGamal.

The ElGamal signature scheme allows that a verifier can confirm the authenticity of a message *m* sent by the signer sent to him over an insecure channel.

Contents

System parameters

- Let *H* be a collision-resistant hash function.
- Let *p* be a large prime such that computing discrete logarithms modulo *p* is difficult.
- Let *g* be a randomly chosen generator of the multiplicative group Z_p^*.

These system parameters may be shared between users.

Key generation

- Choose randomly a secret key *x* with $1 < \mathrm{x} < \mathrm{p} - 1$.
- Compute $\mathrm{y} = \mathrm{g}^{\mathrm{x}} (\mathrm{mod}\ \mathrm{p})$.
- The public key is (*p*,*g*,*y*).
- The secret key is *x*.

These steps are performed once by the signer.

Signature generation

To sign a message *m* the signer performs the following steps.

- Choose a random *k* such that $0 < \mathrm{k} < \mathrm{p} - 1$ and $\gcd(\mathrm{k}, \mathrm{p} - 1) = 1$.
- Compute $r \equiv g^k \pmod{p}$.

- Compute $s \equiv (H(m) - xr)k^{-1} \pmod{p-1}$.
- If s = 0 start over again.

Then the pair *(r,s)* is the digital of *m*. The signer repeats these steps for every signature.

Verification

A signature *(r,s)* of a message *m* is verified as follows.

- $0 < r < p$ and $0 < s < p - 1$.
- $g^{H(m)} \equiv y^r r^s \pmod{p}$.

The verifier accepts a signature if all conditions are satisfied and rejects it otherwise.

Correctness

The algorithm is correct in the sense that a signature generated with the signing algorithm will always be accepted by the verifier.

The signature generation implies

$$H(m) \equiv xr + sk \pmod{p-1}.$$

Hence Fermat's little theorem implies

$$\begin{aligned} g^{H(m)} &\equiv g^{xr} g^{ks} \\ &\equiv (g^x)^r (g^k)^s \\ &\equiv (y)^r (r)^s \pmod{p}. \end{aligned}$$

Security

A third party can forge signatures either by finding the signer's secret key *x* or by finding collisions in the hash function $H(m) \equiv H(M) \pmod{p-1}$. Both problems are believed to be difficult.

The signer must be careful to choose a different *k* uniformly at random for each signature and make sure that *k* or even partial information about *k* is not leaked. Otherwise a third party may be able to deduce the secret key *x* with less difficulty. In particular, if two messages are sent using the same value of *k* then a third party can compute *x*.

See also

- Digital Signature Algorithm
- Elliptic Curve DSA
- ElGamal encryption