The Elgamal Digital Signature Scheme

Key Generation for Elgamal Digital Signature

1. Choose a large prime p.
2. Choose a primitive element α of $\mathbb{Z}_p^*$ or a subgroup of $\mathbb{Z}_p^*$.
3. Choose a random integer $d \in \{2, 3, \ldots, p-2\}$.
4. Compute $\beta = \alpha^d \bmod p$.

The public key is now formed by $k_{pub} = (p, \alpha, \beta)$, and the private key by $k_{pr} = d$.

Elgamal Signature Generation

1. Choose a random ephemeral key $k_E \in \{0, 1, 2, \ldots, p-2\}$ such that $\gcd(k_E, p-1) = 1$.
2. Compute the signature parameters:

$$r \equiv \alpha^{k_E} \bmod p,$$
$$s \equiv (x - d \cdot r)\, k_E^{-1} \bmod p-1.$$

Elgamal Signature Verification

1. Compute the value

$$t \equiv \beta^r \cdot r^s \bmod p$$

2. The verification follows from:

$$t \begin{cases} \equiv \alpha^x \bmod p & \Longrightarrow \text{valid signature} \\ \not\equiv \alpha^x \bmod p & \Longrightarrow \text{invalid signature} \end{cases}$$

Alice **Bob**

1. choose $p = 29$
2. choose $\alpha = 2$
3. choose $d = 12$
4. $\beta = \alpha^d \equiv 7 \bmod 29$

$$\xleftarrow{(p,\alpha,\beta)=(29,2,7)}$$

compute signature for message $x = 26$:

choose $k_E = 5$, note that $\gcd(5,28) = 1$

$r = \alpha^{k_E} \equiv 2^5 \equiv 3 \bmod 29$

$s = (x - d\,r)\,k_E^{-1} \equiv (-10) \cdot 17 \equiv 26 \bmod 28$

$$\xleftarrow{(x,(r,s))=(26,(3,26))}$$

verify:

$t = \beta^r \cdot r^s \equiv 7^3 \cdot 3^{26} \equiv 22 \bmod 29$

$\alpha^x \equiv 2^{26} \equiv 22 \bmod 29$

$t \equiv \alpha^x \bmod 29 \Longrightarrow$ valid signature

Existential Forgery Attack Against Elgamal Digital Signature

Alice **Oscar** **Bob**

$k_{pr} = d$
$k_{pub} = (p, \alpha, \beta)$

$\xleftarrow{(p,\alpha,\beta)}$ $\xleftarrow{(p,\alpha,\beta)}$

1. select integers $i,\ j$
   where $\gcd(j, p-1) = 1$
2. compute signature:
   $r \equiv \alpha^i \beta^j \bmod p$
   $s \equiv -r\,j^{-1} \bmod p -$
1
3. compute message:
   $x \equiv s\,i \bmod p-1$

$\xleftarrow{(x,(r,s))}$

verification:
$t \equiv \beta^r \cdot r^s \bmod p$
since $t \equiv \alpha^x \bmod p$:
valid signature!

The attack is not possible if the message is hashed, which is, in practice, very often the case. Rather than using the message directly for computing the signature, one applies a hash function to the message prior to signing, i.e., the signing equation becomes:

$$s \equiv (h(x) - d \cdot r)k_E^{-1} \bmod p - 1.$$