ElGamal encryption and signature

Tanja Lange

Eindhoven University of Technology

2WF80: Introduction to Cryptology

ElGamal encryption

For historical purposes only

- This scheme does encrypt messages, requires messages to be in G.
- Alice publishes long-term public key $h_A = g^a$,
  keeps long-term private key a.
- Any user can encrypt to Alice using this key:
  - Pick random k, compute $r = g^k$.
  - Encrypt $m \in G$ as $c = (g^a)^k \cdot m$.
  - Send (r, c).
  - Alice decrypts, by computing
    $m = c/(r^a) = (g^a)^k \cdot m/g^{ak}$.
- Positives:
  - Is homomorphic.
  - Is randomized.
- Downsides:
  - Requires $m \in G$.
  - Is homomorphic.
  - Not OW-CCA II secure.

ElGamal signature

- This requires computing inverses modulo the order of g.
  Easiest to describe if $\mathrm{ord}(g) = \ell$ is prime.
- Alice publishes long-term public key $h_A = g^a$,
  keeps long-term private key a.
- Alice signs message m:
  - Pick random k, compute $r = g^k$, and
    $s \equiv k^{-1}(H(m) - ar) \bmod \ell$.
  - Signature is (r, s).
- Anybody can verify signature:
  Compute $g^{H(m)} - r^s \cdot (h_A)^r$, accept if 0.
- Valid signatures get accepted:

ElGamal signature

- This requires computing inverses modulo the order of g.
  Easiest to describe if $\mathrm{ord}(g) = \ell$ is prime.
- Alice publishes long-term public key $h_A = g^a$,
  keeps long-term private key a.
- Alice signs message m:
  - Pick random k, compute $r = g^k$, and
    $s \equiv k^{-1}(H(m) - ar) \bmod \ell$.
  - Signature is (r, s).
- Anybody can verify signature:
  Compute $g^{H(m)} - r^s \cdot (h_A)^r$, accept if 0.
- Valid signatures get accepted:

$$r^s \cdot (h_A)^r = g^{k \cdot k^{-1}(H(m) - ar)} \cdot g^{ar} = g^{H(m)}.$$

  Thus the difference is 0.
  Note that computations in the exponent of g happen modulo the order of g.