4.5 ElGamal Cipher—Idea

The ElGamal cipher is an asymmetric cipher—or more precisely a hybrid cipher—that also relies on the complexity of the discrete logarithm.

The basic public parameters are a prime p and an element $g \in [2 \ldots p-2]$. The order of g in $\mathbb{F}_p^\times$ should be high, preferably g should be a primitive element $\bmod p$.

> p and g may be shared by all participants but also may be individually chosen.

Each participant chooses a random integer

$$d \in [2 \ldots p-2]$$

as private key, und computes

$$e = g^d \bmod p$$

as corresponding public key. Computing d from e is computing a discrete logarithm, hence presumably hard.

The definitioon of the cipher needs one more idea: How to transform a message a in such a way that it can be reconstructed only with knowledge of d?

> The naive idea of sending $e^a = g^{da} \bmod p$ is useless—knowing d doesn't help with decrypting a. Also sending $r = g^a \bmod p$ is useless—the receiver can compute $r^d = e^a \bmod p$ but not a.

The idea is to first generate a message key to be used with a hybrid procedure:

- Alice chooses a random $k \in [2 \ldots p-2]$. As key she will use $K = e^k \bmod p$ where e is the Bob's public key, thus Alice can compute K.
- To share the key K with Bob Alice sends the *key information* $r = g^k \bmod p$ together with the encrypted message.
- Bob computes $r^d = g^{kd} = e^k = K \bmod p$ using his private key d.

As symmetric component of the hybrid encryption the shift cipher in $\mathbb{F}_p^\times$ is used with K as one-time key. So Alice has to generate a new key K for each plaintext block and to send the corresponding key information, doubling the length of the message.

Thus, after generating the key K and the key information r:

- the formula for encryption is $c = Ka \bmod p$,
- and the message to be sent is (c, r).

Bob computes the key K from r, and then decrypts

- $a = K^{-1}c \bmod p$ by congruence division.