Problem Q-2. Group signatures

In class we described the concept of *key escrow* for public key encryption schemes. A user chooser her public key E and matching secret key $D = E^{-1}$. She makes E public and shares the secret key D among a set of trustees, so that if needed they can decrypt her messages by putting together their shares.

A related concept is the concept of **group signatures**. Suppose that a corporation has a public key E used to sign official documents. So a document M will be signed by publishing $\sigma = D(M)$ and such signature can be publicly verified by everybody against the public key E by checking

$$E(\sigma) = M$$

However in order for documents to be signed on behalf of the corporation *all* n of the corporation's vice-presidents (VP's) have to agree. One possible way to achieve this is to give a public key to each VP and have each of them sign the document. Only documents with all n signatures will be accepted as "signed." This is however cumbersome since it involves appending and verifying n signatures for each document.

Group signatures solve the problem by sharing the secret key D among all the VP's. Let D_i be the share of the secret key held by VP i. At signing time each VP produces a *partial signature* $\sigma_i = D_i(M)$ and then these signatures are combined in order to produce $D(M)$.

Example: Assume the encryption scheme is RSA and the public key is (N, e) and the secret key is d. Then a signature of a message M is $\sigma = M^d \bmod N$. It is verified by checking that $\sigma^e = M \bmod N$. One way to create a group RSA signature is to have a trusted party generate the keys (N, e, d), make (N, e) public and give privately to VP i a value d_i such that

$$d = d_1 + d_2 + \ldots d_n \bmod \phi(N)$$

Then the trusted party self–destructs. Suppose now that the VP's want to sign a message M. Each VP produces $\sigma_i = M^{d_i} \bmod N$ and the signature is simply set to

$$\sigma = \prod_{i=1}^{n} \sigma_i = \prod_{i=1}^{n} M^{d_i} \bmod N = M^{\sum_{i=1}^{n} d_i \bmod \phi(N)} = M^d$$

which is the correct corporation signature. Notice that it is necessary for the trusted party to self–destruct otherwise the information contained in it would allow someone to sign messages for the corporation without the help of the VP's.

In this problem you will be asked to come up with a group signature scheme for the following signature algorithm. The algorithm is a modification of ElGamal devised by the teaching staff in order to make this problem solvable. So there is no guarantee that such scheme is secure.

The signature scheme you have to use for the problem is as follows. Let p be a large prime, g a generator of Z_p^*, x the secret key and $y = g^x$ the public key.

When the signer wants to sign a message M she chooses a random number k (modulo $p-1$) and computes

$$a = g^k \bmod p$$

then she computes

$$b = xa + k - M \bmod (p-1)$$

The signature is the pair (a, b). In order to verify the signature one has to check that

$$g^{M+b} = y^a * a \bmod p$$

Notice that as in the ElGamal scheme the value k has to remain secret, otherwise the secret x would be easily computable from a signature.

(a) Give a group signature scheme for the signature algorithm described above. As in the example you can assume the existence of a trusted party.

(b) For the signature scheme described above a trusted party is not needed. Modify your solution to part **(a)** so that it does not use a trusted party for the generation of the shares.