Assignment 2

The following is the ElGamal encryption scheme (with a tiny difference) in Assignment 1 (over Group 14 of RFC 3526):

Key Generation

1. Set p as the prime in Group 14 of RFC 3526, set q=(p-1)/2 as the order of group G
2. Set g=2 as the generator of G
3. Sample s from $\mathrm{Z_q}$ as the secret key
4. Set $\mathrm{u} = \mathrm{g^s}$ from Group G as the public key
5.

Encryption(u, b)

1. On input public key u, (public parameters p, q), and message b ∈ $[0, \mathrm{B}]$ for some integer B, sample randomness β from $\mathrm{Z_q}$
2. Compute and return ciphertext $(\mathrm{c_1}, \mathrm{c_2}) = (\mathrm{g}^{\beta}, \mathrm{u}^{\beta} \cdot \mathrm{g}^{b})$

Decryption(s, $(\mathrm{c_1}, \mathrm{c_2})$)

1. On receiving $(\mathrm{c_1}, \mathrm{c_2})$, compute $\mathrm{t} = \mathrm{c_2}/c_1^s$.
2. Find b such that $\mathrm{t} = \mathrm{g}^{b}$.

Remark. In assignment 1, we set the message space as G, and encrypt message b from the group G as $(\mathrm{c_1}, \mathrm{c_2}) = (\mathrm{g}^{\beta}, \mathrm{u}^{\beta} \cdot \mathrm{b})$. Here we chose a small integer B, set the message space as [0, B], and encrypt message b from [0, B] as $(\mathrm{c_1}, \mathrm{c_2}) = (\mathrm{g}^{\beta}, \mathrm{u}^{\beta} \cdot \mathrm{g^b})$.

Generally, it is hard to compute b given $\mathrm{t} = \mathrm{g}^{b}$, if b is large and random (this is the discrete logarithm problem). However, since we require B to be small (10 for example), we can efficiently find b.

In this assignment, please generate

Task 1: two ElGamal encryptions of 0 or 1 and generate a SIGMA proof that they are indeed encryptions of 0 or 1.

Task 2: an ElGamal encryption of b from [0, 7] and generate a SIGMA proof that it is indeed an encryption of b from [0, 7].

Refer to Appendix A for SIGMA proof for DDH relation. The following are the concrete requirement.

Task 1

- Design a SIGMA proof that two ElGamal ciphertexts are encryptions of 0 or 1.
- Implement the ElGmamal encrytions, and the SIGMA protocol in Sege

Hint: First, to show that $(\mathrm{c_1}, \mathrm{c_2})$ is an encryption of 0 or 1, we just need a proof that $(g,\ u,\ \mathrm{c_1}, \mathrm{c_2})$ or $(g,\ u,\ \mathrm{c_1}, \mathrm{c_2}/g)$ is DDH tuple, which is an OR composition of $(g,\ u,\ \mathrm{c_1}, \mathrm{c_2})$ is DDH or $(g,\ u,\ \mathrm{c_1}, \mathrm{c_2}/g)$ is DDH. Refer to [BS2, Figure 20.1]

Second, to show that two ciphetexts are the encryption of 0 or 1. We apply the AND-composition, in which the two SIGMA can share the challenge.

Please refer to [BS23, section 19.7] on details of AND, OR composition of two SIGMA protocols.

Task 2

- Design a SIGMA proof that the message of an ElGamal ciphertext is in [0, 7]
- Implement the Elgmamal encrption, and the SIGMA protocol in Sege

Hint: Prove that (c_1, c_2) is an encryption of 0, 1, 2, 3, 4, 5, 6 or 7 using OR composition.

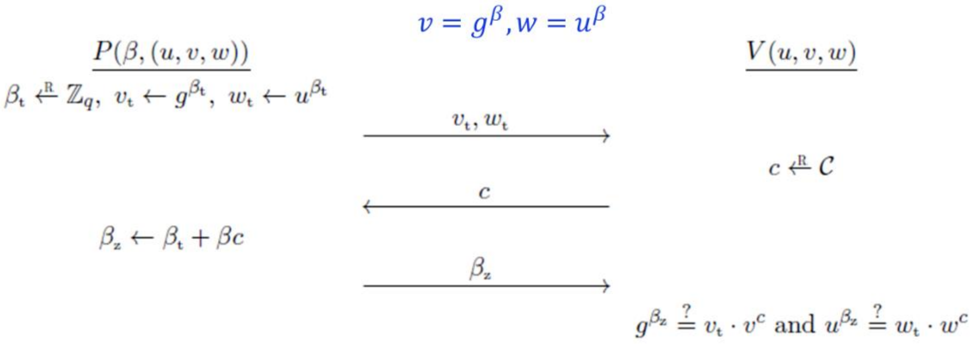
Please refer to [BS23, section 19.7] on details of OR composition of SIGMA protocols.

Requirement: You need to submit a detailed lab report, with screenshots, to describe what you have done and what you have observed. You also need to provide an explanation for the observations that are interesting or surprising. Please also submit the code followed by an explanation. Simply attaching code without any explanation will not receive credits.

Reference:

[BS23] https://crypto.stanford.edu/~dabo/cryptobook/BonehShoup_0_6.pdf

Appendix A: A SIGMA proof on $(g, u, v, \ w)$ is an DDH-tuple


$v = g^{\beta}, w = u^{\beta}$
$P(\beta, (u, v, w))$
$V(u, v, w)$
$\beta_t \xleftarrow{R} \mathbb{Z}_q,\ v_t \leftarrow g^{\beta_t},\ w_t \leftarrow u^{\beta_t}$
v_t, w_t
$c \xleftarrow{R} \mathcal{C}$
c
$\beta_z \leftarrow \beta_t + \beta c$
β_z
$g^{\beta_z} \stackrel{?}{=} v_t \cdot v^c$ and $u^{\beta_z} \stackrel{?}{=} w_t \cdot w^c$

Please refer to [BS23, section 19.5.2] for a full description.

Appendix B: SageMath is a free open-source mathematics software system licensed under the GPL. It builds on top of many existing open-source packages. You may install SageMath on your laptop by following guide provided by https://www.sagemath.org/ or just run your code using online project https://sagecell.sagemath.org/