Homework 2

1 Problem 1 (30 points)

(a) Suppose you are given a group $\mathbb{G}$ of *unknown order* p, along with a generator g. The only thing you know about $\mathbb{G}$ is that it's order is between 2^λ and $2^{\lambda+1}$. Assume decisional Diffie-Hellman is still hard on $\mathbb{G}$. Show how to build a non-interactive key agreement protocol using $\mathbb{G}$. This will require tweaking the usual Diffie-Hellman protocol.

(b) Back to the known-order setting, suppose decisional Diffie-Hellman is *easy* on $\mathbb{G}$ (perhaps, because there is an efficiently computable pairing), but that *computational* Diffie-Hellman remains hard. Explain why the the Diffie-Hellman key exchange protocol discussed in class no longer yields a pseudorandom key.

(c) Explain how to tweak the protocol to yield a pseudorandom key. Your scheme should:

- Support keys of length at least λ.
- Remain non-interactive: there is a single message from Alice to Bob and from Bob to Alice, and both messages are sent simultaneously.
- Before the protocol begins, the only information that Alice and Bob share is the group $\mathbb{G}$ and a generator g.

Remember to prove the security of your protocol.

Hint: for part (c), it will be useful to have the following strengthening of the Goldreich-Levin theorem. Let S be a PPT algorithm that takes as input the security parameter, and samples pairs (s, aux). We say that S is computationally unpredictable if, for all PPT A,

$$\Pr[A(\mathsf{aux}) = s : (\mathsf{aux}, s) \leftarrow S(1^\lambda)] < \mathsf{negl}(\lambda)$$

Let $n(\lambda)$ be the length of s outputted by S. Then for any computationally unpredictable S, the following two distributions are computationally indistinguishable:

$$(r, \mathsf{aux}, < r, s >) : (\mathsf{aux}, s) \leftarrow S(1^\lambda), r \leftarrow \{0,1\}^{n(\lambda)} \text{ and}$$
$$(r, \mathsf{aux}, b) : (\mathsf{aux}, s) \leftarrow S(1^\lambda), r \leftarrow \{0,1\}^{n(\lambda)}, b \leftarrow \{0,1\}$$

2 Problem 2 (40 points)

For most elliptic curves, the pairing operation unfortunately does work as cleanly as we discussed in class. Concretely, the source group will be $\mathbb{G}'$, which is (inefficiently) isomorphic to $\mathbb{Z}_p \times \mathbb{Z}_p$. In particular, $\mathbb{G}'$ is *not* cyclic. Then the pairing has the following features:

- $e : \mathbb{G}' \times \mathbb{G}' \to \mathbb{G}_T$, where $\mathbb{G}_T$ is some cyclic group of order p (concretely, it is a subgroup of the multiplicative group of an appropriate finite field).
- $e(g, h) = e(h, g)^{-1}$. That is, the pairing is *anti-symmetric*.
- e is bilinear ($e(g_1 \times g_2, h) = e(g_1, h) \times e(g_2, h)$ and $e(g, h_1 \times h_2) = e(g, h_1) \times e(g, h_2)$)
- e is *non-degenerate* ($e(g, h)$ is not identically 0)

(a) One attempt to construct a pairing of the form we discussed in class is to simply choose a random $g \in \mathbb{G}'$, and let $\mathbb{G}$ be the group generated by g. We then let e simply be the restriction of e to $\mathbb{G}$. Unfortunately, this will not work. Show that the pairing, when restricted to $\mathbb{G}$, is useless.

(b) The solution, usually, is to define *two* source groups. Specifically, let $g, h \in \mathbb{G}'$ be random in $\mathbb{G}'$, and let $\mathbb{G}_1$ be the group generated by g, and $\mathbb{G}_2$ the group generated by h. Then we can consider the pairing e as the restriction to $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$.

Show that, with overwhelming probability, e will be non-generate.

(c) The resulting pairing is known as an *asymmetric* pairing, whereas the kind we discussed in class with $\mathbb{G}_1 = \mathbb{G}_2$ is known as a *symmetric* pairing. Working with asymmetric pairings is a little different that asymmetric, as we will now discuss.

Explain, intuitively, why the decisional Diffie-Hellman problem might remain hard on an asymmetric pairing (whereas we know it is easy on a symmetric pairing).

(d) Explain why the 3-party key non-interactive agreement protocol we discussed in class is no longer correct on a asymmetric pairing. Formulate a new 3-party non-interactive key agreement protocol that is correct on asymmetric pairings. You do not need to prove security, but informally argue why there are no trivial attacks on your scheme

For those who are interested, some notes:

- In general, $\mathbb{G}_1, \mathbb{G}_2$ are not chosen by choosing random generators, but are chosen rather specifically for practical considerations.
- Symmetric pairings are known, from a special class of elliptic curves called supersingular curves. Unfortunately, supersingular curves are subject to sub-exponential attacks based on the MOV attack discussed in class. As a result, parameters have to be set somewhat higher, resulting in less efficient schemes.
- Symmetric pairings are often referred to as "Type 1". The asymmetric pairing discussed above is often referred to as "Type 3". A "Type 2" pairing is a symmetric pairing that additionally has an efficient homomorphism $\psi : \mathbb{G}_2 \to \mathbb{G}_1$. This homomorphism resurrects some (but not all) of the features of a Type 1 pairing.

3 Problem 4 (30 points)

In this problem, you will show how to sample from the discrete Gaussian distribution $D_{\sigma,c}$. You are given the following fact:

Theorem 1 *Suppose $\sigma \geq 1$. Let t be a function that grows faster than $\sqrt{\log \lambda}$. Then there is a negligible function* negl *such that*

$$\Pr[|x - c| > \sigma t(\lambda) : x \leftarrow D_{\sigma,c}] < \mathsf{negl}(\lambda)$$

Also, we will assume access to a procedure that samples uniformly random real numbers between 0 and 1. We will not worry about the precision of real numbers; assume we can compute and store infinitely precise numbers.

Let $\rho_\sigma(y) = e^{-\pi y^2/\sigma^2}$. Notice that $0 \leq \rho_{\sigma,c} \leq 1$. Let $p_{\sigma,c}(x) = \rho_\sigma(x-c)/\sum_{z=-\infty}^{\infty} \rho_\sigma(z-c)$. Then $p_{\sigma,c}(x) = \Pr[x : x \leftarrow D_{\sigma,c}]$.

We will use rejection sampling. One approach is to choose a random integer x, and then with probability $p_{\sigma,c}(x)$, accept and output x. Otherwise, throw away x and repeat from the beginning. Notice that in each iteration, any x is outputted with probability proportional to $p_{\sigma,c}(x)$. Therefore, once the algorithm terminates, the distribution of outputs is exactly $D_{\sigma,c}$.

Unfortunately, the above does not quite work for two reasons:

- It is not possible to sample a uniformly random x over all integers (since the expected length of a random integer is infinite)
- Suppose σ is quite large (say, exponential). Then $p_{\sigma,c}(x)$ is exponentially small for all x. In this case, the procedure above will take an exponential number of iterations to terminate, and is therefore inefficient.

Show how to fix the above problems and give a protocol that terminates in an expected polynomial number of iterations, and outputs a sample x from the discrete Gaussian. The algorithm may output a distribution that is slightly different from the discrete Gaussian, but it should be negligibly close (in the parameter λ). You may assume that $\sigma \geq 1$. Prove that the number of iterations is bounded by a polynomial in λ.