Cryptography Exercise Sheet 10

Prof. Dr. Thomas Wilke, Dr.-Ing. Kim-Manuel Klein

29. Mai 2020

1. **Problem:** *Computationally unbounded adversaries (Exercise 10.1 in BS)*

   Show that an anonymous key exchange protocol P (as in Definition 10.1) cannot be secure against a computationally unbounded adversary. This explains why all protocols in this chapter must rely on computational assumptions.

2. **Problem:** *Random self-reduction for CDH (I) (Exercise 10.4 in BS)*

   Consider a specific cyclic group $\mathbb{G}$ of prime order q generated by $g \in \mathbb{G}$. For $u = g^\alpha \in \mathbb{G}$ and $v = g^\beta \in \mathbb{G}$, define $[u, v] = g^{\alpha\beta}$, which is the solution instance (u, v) of the CDH problem. Consider the randomized mapping from $\mathbb{G}^2$ to $\mathbb{G}^2$ that sends (u, v) to $(\tilde{u}, v)$, where

   $$p \xleftarrow{R} \mathbb{Z}_q,\ \tilde{u} \leftarrow g^p u.$$

   Show that

   (a) $\tilde{u}$ is uniformly distributed over $\mathbb{G}$;

   (b) $[\tilde{u}, v] = [u, v] \cdot v^p$.

3. **Problem:** *Problems equivalent to CDH (Exercise 10.15/10.16 in BS)*

   Consider a specific cyclic group $\mathbb{G}$ of prime order q generated by $g \in \mathbb{G}$. Show that the following problems are deterministic poly-time equivalent:

   (a) Given g^α and g^β, compute $g^{\alpha\beta}$ (this is just the Computational Diffie-Hellman problem).

   (b) Given g^α, compute $g^{(\alpha^2)}$.

   (c) Given g^α with $\alpha \neq 0$, compute $g^{1/\alpha}$.

   (d) Given g^α and g^β with $\beta \neq 0$, compute $g^{\alpha/\beta}$.

   Note that all problem instances are defined with respect to the same group $\mathbb{G}$ and generator $g \in \mathbb{G}$.

4. **Problem:** *A proper trapdoor permutation scheme based on RSA (Exercise 10.24/10.25 in BS)*

   As discussed in Section 10.3, our RSA-based trapdoor permutation scheme does not quite satisfy our definitions, simply because the domain on which it acts varies with the public key. This exercise shows one way to patch things up. Let ℓ and e be parameters used for RSA key generation, and let G be the key generation algorithm, which outputs a pair (pk, sk). Recall that $pk = (n, e)$, where n is an RSA modulus, which is the product of two ℓ-bit primes, and e is the encryption exponent. The secret key is $sk = (n, d)$, where d is the decryption exponent corresponding to the encryption exponent e.

Choose a parameter L that is a substantially larger than 2ℓ, so that $n/2^L$ is negligible. Let $\mathcal{X}$ be the set of integers in the range $[0, 2^L)$. We shall present a trapdoor permutation scheme (G, F^*, I^*), defined over $\mathcal{X}$. The function F^* takes two inputs: a public key pk as above and an integer $x \in \mathcal{X}$, and outputs an integer $y \in \mathcal{X}$, computed as follows. Divide x by n to obtain the integer quotient Q and remainder R, so that $x = nQ + R$ and $0 \leq R < n$. If $Q > 2^L/n - 1$, then set $S := R$; otherwise, set $S := R^e \bmod n$. Finally, set $y := nQ + S$.

(a) Show that $F^*(pk, \cdot)$ is a permutation on $\mathcal{X}$, and give an efficient inversion function I^* that satisfies $I^*(sk, F^*(pk, x)) = x$ for all $x \in \mathcal{X}$.

(b) Show under the RSA assumption, (G, F^*, I^*) is one-way.