6 General One-Way and Trapdoor Functions

In this section, we will try to generalize what we've seen so far. For example, we know how to build a secure encryption out of RSA, but what exactly is RSA itself? In modern terms, it is a trapdoor permutation family, which we define below.

6.1 One-Way Functions

Let us first introduce one-way functions. We've actually seen concrete examples of them before; this is just a generalization, so we can talk of a one-way function f independent of its particular implementation.

Definition 1. A function $f : \{0,1\}^* \rightarrow \{0,1\}^*$ is one-way if

1. it is polynomial-time computable;
2. it is hard to invert, i.e., for all probabilistic polynomial-time A there exists a negligible function η such that, for all k, $\Pr[f(A(f(x), 1^k)) = f(x)] \leq \eta(k)$, where the probability is taken over a random choice of k-bit string x and coin tosses of A.

Note that it's important that we are not requiring A to find x; rather, any inverse of $f(x)$ is fine. Of course, if f is a permutation (i.e., a bijective function), then it would be equivalent to require A to find x, because x is the only inverse of $f(x)$.

Note also the importance of selecting the input to A: the input is not selected uniformly at random; rather, x is selected uniformly at random, and the input is $f(x)$. Of course, again, if f is a permutation, then the two are equivalent.

An example is the following f: split the k-bit input into strings a of length $\lfloor k/2 \rfloor$ and b of length $\lceil k/2 \rceil$, and output $c = ab$. The inverter A would have to find two *large* factors of c, which is believed to be hard. Note that the input c of A is not a uniformly selected integer; in particular, we know that it has two factors of (nearly) the same length.

The existence of one-way functions is the minimal assumption necessary (though often not sufficient) for almost anything interesting in cryptography. Note that the assumption that one-way functions exist is stronger than the assumption that P≠NP (intuitively, because one-way functions are hard on the average case, where as it could be that NP-complete problems are hard only very infrequently).

A *one-way permutation* is a one-way function that is a bijection of $\{0,1\}^k$ to $\{0,1\}^k$ for each k.

6.2 One-Way Function Families

The examples we've seen in class, such as modular squaring, RSA, and Discrete Logarithm, are not quite one-way functions by the above definition. Rather, they are one-way function families, as defined below.

Definition 2. Let I be an index set. A collection of functions $\{f_i : D_i \rightarrow R_i\}_{i \in I}$ is called one-way, if:

1. there exists a probabilistic polynomial-time algorithm Gen that, on input 1^k, picks $i \in I$;
2. there exists a probabilistic polynomial-time algorithm M that, on input $i \in I$, picks $x \in D_i$;
3. given i and x, the value $f_i(x)$ is polynomial-time computable;
4. for all probabilistic polynomial-time A there exists a negligible function η such that, for all k, if i is chosen by $\mathrm{Gen}(1^k)$ and x is chosen by $M(i)$, $\Pr[f_i(A(f_i(x), i, 1^k)) = f_i(x)] \leq \eta(k)$, where the probability is taken over coin tosses of Gen, M and A.

For example, for Discrete Logarithm, the index set $I = \{(p, g)|p$ is prime $, g$ is a generator of $\mathbb{Z}_p^*\}$, and for $(p, g) \in I$, $D_{(p,g)} = R_{(p,g)} = \mathbb{Z}_p^*$ and $f_{(p,g)}(x) = g^x \bmod p$.

A *collection of one-way permutations* is a collection of one-way functions with the additional property that f_i is a permutation. The discrete logarithm collection is actually a collection of one-way permutations.

6.3 Trapdoor Permutations

A collection of one-way permutations with the additional property that the (unique) inverse is easy to obtain with some special information is called a collection of trapdoor permutations.

Definition 3. A collection of one-way permutations $\{f_i : D_i \to R_i\}_{i \in I}$ is called *trapdoor* if there exists a probabilistic polynomial-time algorithm Inv and if Gen, in addition to outputting $i_\in I$, outputs a value t with the following property: for all $x \in D_i$, $\text{Inv}(t, f_i(x)) = x$.

For example, RSA is a collection of trapdoor permutations. The index set consists of pairs (n, e); the trapdoor information t is (n, d); and the domain and the range are $\mathbb{Z}_n^*$.

6.4 Generalizing Results

To obtain a pseudorandom generator, both the Blum-Micali and the Blum-Blum-Shub generators simply selected a one-way permutation from a family, and iterated it multiple times on a random initial seed, each time outputting a bit that's hard to predict. It is natural to ask whether for any one-way permutation (family) there is such a bit. The following theorem of Goldreich and Levin answers this question in the affirmative. We state it somewhat informally, and do not prove it here.

Theorem 1 ([GL89]). *Let f be a one-way function (the same also holds for families of one-way functions). Let r be a random k-bit value. Then, for a random k-bit x, the bit $r \cdot x$ is hard to compute with probability greater than $1/2$, given $f(x)$ and r. (Here $r \cdot x = r_1x_1 \oplus r_2x_2 \oplus \dots r_kx_k$, the inner-product modulo 2 of r and x.)*

Therefore, our constructions of pseudorandom generators extend to *any* one-way permutation f (and, similarly, one-way permutation family). We simply take our seed to be (x, r), let $x_0 = x, x_i = f(x_{i-1})$, and output the bits $b_i = x_i \cdot r$.

Hence, we get

Theorem 2. *If one-way permutations (or families) exist, then so do pseudorandom generators.*

However, one-way functions are a weaker assumption, and it would be nice to know if pseudorandom generators can be based on just one-way functions, not permutations. The following theorem of Håstad, Impagliazzo, Levin and Luby shows that one-way functions suffice. It is quite difficult to prove.

Theorem 3 ([HILL99]). *Pseudorandom generators exist if and only of one-way functions exist.*

Thus, one-way functions suffice for symmetric encryption. However, they do not suffice for public-key encryption: you really need the trapdoor to be able to go back. Note also that by generalizing our previous two bit-by-bit constructions, we know that trapdoor permutations suffice.

Finally, I want to mention two constructions of Levin's [Lev87, Lev03] that address the existence of one-way functions. In both, he constructs a single function U with the following property: U is one-way if one-way functions exist. U is known as the *universal one-way function*. The question of whether one-way functions exist reduces to the question of whether this specific single function is one-way.

References

[GL89] O. Goldreich and L. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing*, pages 25–32, Seattle, Washington, 15–17 May 1989.

[HILL99] J. Håstad, R. Impagliazzo, L.A. Levin, and M. Luby. Construction of pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[Lev87] Leonid A. Levin. One-way functions and pseudorandom generators. *Combinatorica*, 7(4):357–363, 1987.

[Lev03] Leonid A. Levin. The tale of one-way functions. *Problems of Information Transmission (Problemy Peredachi Informatsii)*, 39(1):92–103, 2003. Available at `http://arxiv.org/abs/cs.CR/0012023`.