Lossy Trapdoor Functions

Giacomo Fenzi

ETH Zurich

22 April 2021

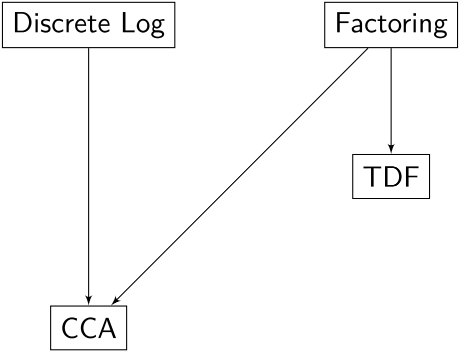
Motivation

- Trapdoor Functions are basic primitive, but hard to instantiate
- CCA Security from factoring and discrete log but not lattices

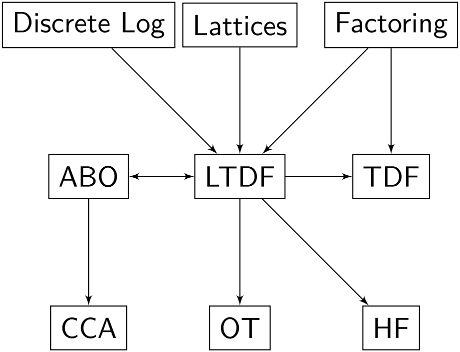
Results

- Introduce Lossy Trapdoor Functions (LTDFs)
- Realize LTDFs from factoring, discrete log *and* lattices
- Show LDTFs imply TDFs
- Black box construction of CCA-secure (witness recovering) cryptosystems, collision-resistant hash functions and oblivious transfer protocols.

Connections



Connections



Notation and Entropy

- λ is the security parameter, and we will abbreviate $n(\lambda) = \text{poly}(\lambda)$ as simply n

Trapdoor Functions

Informally, a trapdoor function is family of functions that are hard to invert without access to some additional information called a trapdoor

Definition

A trapdoor function consists of three PPT algorithms (S, F, F^{-1}) such that:

- *Easy to sample and invert with trapdoor.* $S(1^\lambda) \to (s, t)$ such that $F(s, -)$ is an injective function on $\{0,1\}^n$ and $F^{-1}(t, -)$ is its inverse
- *Hard to invert without.* For *any* PPT inverter $\mathcal{A}$ we have that $\mathcal{A}(1^\lambda, s, F(s, x))$ outputs x with negligible probability.

Example of Trapdoor

RSA Encryption! In trapdoor form:

- $S(1^\lambda)$ generates N, e, d as in RSA, set $s = (N, e)$ and $t = (d)$ and returns (s, t)
- $F(s, x)$ computes $x^e \mod N$
- $F^{-1}(t, c)$ computes $c^d \mod N$