UMass CS Theory Seminar

Lectures by Various
Scribed by Ben Burns

UMass Amherst

Spring 2022

Contents

1 Post Quantum Crypography - Katerina Sotiraki

Supersingular elliptic curve isogeny

Lattices are geometric structures, represented using matrices

Collision-Resistant Hashing: compressing functions where it is computationally hard to find collisions

Example (Digital signatures)**.** How can we be sure that the document or application we are opening is that we are looking for, by verifying the signature. The hardness of finding collisions is important, because otherwise you can find other files that hash in the same way, and therefore are recognized as signed

Example (Blockchains)**.** Each block contains a hash of the previous block. If it is not hard, a malicious agent can insert a malicious block that is recognized as valid

Is there an optimal collision-resistant hash function? i.e is there a best among factoring, RSA.

Start with a hard problem. What if we start an NP-complete problem like SAT

Cryptographic hardness is different than computational hardness. In other words, average-case hardness is not the same as worst-case hardness. Hash functions always have collisions, but for NP problems, we don't know if there is a solution and are attempting to verify if there is one.

Instead, start with lattices. Average-case hardness ~ worst-case hardness. The second property is our result.

Starting with all collision-resistent hash functions, we can recognize a family that is the hardest.

Theorem 1 (S Zirdelis FOCS '18)**.** *There is an optimal construction for worst-case collision-resistant hash function family based on a lattice problem*

Definition 2 (Total Search problem)**.** A search problem where a solution will always exist

For example, integer factorization will always have a solution by the Fundamental Theorem of Arithmetic

Funding collisions in compressing functions is total. Compressing function: $f(x_1, x_2) = x_1 \wedge x_2$, must have a collision by pigeonhole principle, since our output space has is strictly smaller than our input space.

[Ajtai '94] Hash functions from lattices. Key is a rXm matrix of $\mathbb{Z}_q$ entries, input is a binary vector of size m, and the output is matrix vector multiplication mod q.

Why is it hard? Short integer solution problem

Binary inverible matrix

Back propagation. Given vector x and matrix g, find prepended matrix y so that yx in kernel of g. We can find y because g is binary invertible.

PPAD complexity

Finding Nash Equilibrium