Notes for Lecture 11

In this lecture, we show a construction of multilinear maps. There are couples of different constructions in the literature [CLT13, GGH13, GGH15]. We are going to talk about the construction of [GGH13] which is the best understood in terms of their security.

So let us first give some notations.

- $\mathbb{Z}$ for integers and $\mathbb{Z}[x]$ for integer polynomials;
- Let $P(x)$ be a monic degree n polynomial where "monic" means that the leading coefficient is 1; i.e. $P(x) = x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$, This polynomial P defines an equivalent class on the polynomial ring $\mathbb{Z}[x]$: we say $Q_0(x)$ is equivalent to $Q_1(x)$ or $Q_0(x) \equiv Q_1(x) \pmod{P(x)}$ iff $p(x) \mid (Q_0(x) - Q_1(x))$, in other words,
$$\exists R(x) \in \mathbb{Z}[X], P(x)R(x) = Q_0(x) - Q_1(x)$$
It is easy to see that it defines an equivalent class.
- $\mathbb{Z}[x]/P(x)$ is the set of equivalent classes defined by $P(x)$ where P partitions $\mathbb{Z}[x]$ into a bunch of equivalent classes.

For C be an equivalent class in $\mathbb{Z}[x]/P(x)$ where $P(x) = x^n + b_{n-1}x^{n-1} \cdots b_1 x + b_0$, we can actually associate it with a vector $x_c \in \mathbb{Z}^n$. Assume $Q(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$[1] is a polynomial in the equivalent class C, let $Q'(x) = Q(x) - a_n P(x)$. We find that

$$Q'(x) = (a_{n-1} - a_n b_{n-1})x^{n-1} + (a_{n-2} - a_n b_{n-2})x^{n-2} + \cdots + (a_0 - a_n b_0)$$

where $Q'(x)$ is a polynomial of degree at most $n-1$. So we can describe this polynomial by the vector $(a_{n-1} - a_n b_{n-1}, a_{n-2} - a_n b_{n-2}, \cdots, a_0 - a_n b_0) \in \mathbb{Z}^n$. And this description is unique because for any $Q'_1(x)$ and $Q'_2(x)$ of degree $n-1$, it is easy to see that $Q'_1(x) \equiv Q'_2(x) \pmod{P(x)}$ if and only if $Q'_1(x) = Q'_2(x)$ (P is of degree n).

We can think $\mathbb{Z}[x]/P(x) \cong \mathbb{Z}^n$. And we can actually define additions and multiplications on $\mathbb{Z}^n$. Addition is just component-wise. But for multiplication, we can first interpret two vectors as polynomials in $\mathbb{Z}[x]$, multiply them in $\mathbb{Z}[x]$ and finally map the result back in $\mathbb{Z}^n$.

[1] Here Q can have arbitrary degree. We can still have the polynomial $Q'(x) = Q(x) - A(x)P(x)$ by doing "long division method".

So we are going to be interested in a particular $P(x) = x^n + 1$ where n is a power of 2. This polynomial $P(x)$ is called **Cyclotomic**. For this class, we will not worry about why this particular $P(x)$ is chosen. Having this polynomial allows module operations to be quite easy and also this $P(x)$ is irreducible. And let $\mathsf{R} = \mathbb{Z}[\mathsf{x}]/\mathsf{P}(\mathsf{x})$. We use $\mathsf{R_q} = \mathsf{R}/\mathsf{qR}$ denote all vectors in R such that each coefficient is mod by q. And we can choose a specific q such that $\mathsf{R_q}$ is a field.

Now we turns to rational polynomials instead of integer polynomials. Let $\mathsf{K} = \mathbb{Q}[\mathsf{x}]/(\mathsf{x^n} + 1) \cong \mathbb{Q}^{\mathsf{n}}$ and K is a field.

Let $g \in \mathsf{R}$ and we define $\langle g \rangle = \{ag : a \in \mathsf{R}\}$. It is easy to see that $\langle g \rangle$ is a (principle) ideal such that

- $(\langle g \rangle, +)$ is a subgroup of $(\mathsf{R}, +)$;
- for all $r \in \mathsf{R}$ and $x \in \langle g \rangle$, we have $r \cdot x, x \cdot r \in \langle g \rangle$;

By giving the ideal $\langle g \rangle$, we can also define the set of equivalent classes $\mathsf{R}/\langle \mathsf{g} \rangle$.

Now we are giving the construction of multilinear maps in [GGH13].

Setup

- Choose a random "short" $g \in \mathsf{R}$ ($g^{-1} \in \mathsf{K}$ is also "short") where "short" means all the coefficients are "short" or in other words, all the coefficients are generated by discrete gaussian.
- We check whether g^{-1} is also short if not go back to step 1. Actually the paper shows that we will not repeat it "too many" times. You will see that why we need "short" property for g^{-1} later.
- Let $\mathsf{QR} = \mathsf{R}/\langle \mathsf{g} \rangle$ to be the plaintext space. it is actually a finite object. We can actually assume it is of prime order with the right choice of g. g is going to be **secret**.
- A "level 0" encoding of an equivalent class $e + \langle g \rangle$ is a "short" description $c \in e + \langle g \rangle$.
- Choose an uniformly random **secret** $z \in \mathsf{R_q}$ (not required to be "short") for encoding at level $i > 0$. A "level i" encoding of $e + \langle g \rangle$ has the form $c/z^i \mod q$ where $c \in e + \langle g \rangle$ is short. We will denote a level i encoding of an element $a \in \mathsf{QR}$ as $[a]_i$.
- The output $\mathsf{params} = \{[1]_1, \text{ other basic parameters}\}$.

With the above parameters, a level i encoding of $a + \langle g \rangle$ is a short $c \in a + \langle g \rangle$ divided by z^i. So $[1]_1$ generated in Setup is actually $\frac{1+gr}{z}$ for right r such that $1+gr$ is "short".

It is easy to see that it satisfies that properties of multilinear maps:

- For two encodings $[a]_i, [b]_i$ at the same level i, $[a]_i + [b]_i = \frac{a+b}{z^i} = [a+b]_i$ where $a + b$ is also "short", i.e. it simply doubles.
- For two encodings $[a]_i, [b]_j$ at different levels i, j, $[a]_i \times [b]_j = \frac{ab}{z^{i+j}} = [a+b]_{i+j}$; assume each entry in a, b is bounded by σ, then each entry in $a \cdot b$ is bounded by $n\sigma^2$; it is also "short".

Here is the sampling procedure at level 0.

Sample(params, 0)

- Sample a random level 0 "short" encoding $c \in \mathsf{R}$ by discrete Gaussian. One can show that c is an encoding of a random $a \in \mathsf{QR}$.
- Output c.

Now let's go to higher levels. Sampling a random encoding at level $i > 0$ is a *secret* procedure.

Sample(params, z, i)

- Sample a random level 0 "short" encoding $c \in \mathsf{R}$ by discrete Gaussian; so this c is an encoding of a random e
- Return $c/z^i \mod q$.

However the scheme is not enough. Consider we are constructing Key Exchange scheme from this "multilinear maps": three players $\mathcal{A}, \mathcal{B}, \mathcal{C}$ uniformly draw $[a]_0, [b]_0, [c]_0$ using Sample procedure and they encode $[a]_1, [b]_1, [c]_1$ by simply computing $[a]_1 = [a]_0 \times [1]_1$ etc. The problem here is that given $[a]_1$, an adversary can simply get $[a]_0$ by doing division $[a]_1/[1]_1$ because $[1]_1$ is public. To solve this problem, we need to introduce a procedure called ReRandomize.

Before describing ReRandomize, we first modify Setup procedure and the output params. The Setup procedure also prepares a set $S = \{x_1, x_2, \cdots, x_m\}$ with size m such that each x_i is an encoding of 0 at level 1, $x_i = gr_i/z = [0]_1$. And let the

output params be $\{[1]_1, S, \text{other basic parameters}\}$. With the set S, we can describe $\mathsf{ReRandomize}$ at level 1: it takes params and an encoding $[a]_1$, and outputs

$$[a]_1 + \sum_{i \in B} x_i \quad \text{where } B \text{ is an uniform chosen subset of } S$$

In the paper, it shows that with a carefully chosen m, the new encoding is indistinguishable from a fresh generated $[a]_1$.

Zero Test

We prepare a variable $\mathsf{P}_{zt} = z^k h/g \mod q$ where h is moderately "short". Also we add this term into params. So we have $\mathsf{params} = \{[1]_1, S, \mathsf{P}_{zt}, \text{other basic parameters}\}$. Then the zero test procedure $\mathsf{ZeroTest}(\mathsf{params}, [a]_k)$ is defined as: compute $\mathsf{P}_{zt}[a]_k \mod q = h(gr + a)/g = hr + ha/g \mod q$. We have two cases:

- In the first case $a = 0$, the output is simply hr. We have already known h is "short". Because gr is short and g^{-1} is "short", r itself is "short". So hr is also "short".
- In the second case $a \neq 0$, then we have the term $ha/g \pmod q$ which is not "short".

But Zero Test is not enough for $\mathsf{Key\ Exchange}$, we need to extract a from $[a]_k$. Intuitively since for any two encodings of a, the difference $\mathsf{P}_{zt}\left(\frac{a+gr_1}{z^k}\right) - \mathsf{P}_{zt}\left(\frac{a+gr_2}{z^k}\right)$ is small (the difference of the two encodings is the encoding of 0 at level k), we claim the high order bits are the same. So we can throw away low order bits and use extractor to get pseudo randomness.

So $\mathsf{params} = \{[1]_1, S, \mathsf{P}_{zt}\}$. For obfuscation purpose, we only need P_{zt} to be public. Finally, this construction can be easily extended to asymmetric multilinear maps by replace z with $z_1, z_2, \cdots, z_k$.

References

[CLT13] Jean-Sébastien Coron, Tancrede Lepoint, and Mehdi Tibouchi. Practical multilinear maps over the integers. In *Advances in Cryptology–CRYPTO 2013*, pages 476–493. Springer, 2013. 1

[GGH13] Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 1–17. Springer, 2013. 1, 2

[GGH15] Craig Gentry, Sergey Gorbunov, and Shai Halevi. Graph-induced multilinear maps from lattices. In *Theory of Cryptography Conference*, pages 498–527. Springer, 2015. 1