Multilinear Maps

Florian Hess (Oldenburg)

Tel Aviv, February 7, 2013

Introduction

The construction of cryptographic multilinear maps has been a long standing open problem.

There is a very recent construction by Garg, Gentry and Halevi based on ideal lattices.

In the following, a brief and (over)simplified overview of this construction will be given.

Multilinear Pairings

Let $G_1, \cdots, G_n$ and G_T be abelian groups.
A non-degenerate multilinear map is a map

$$e : G_1 \times \cdots \times G_n \to G_T$$

satisfying the following conditions.

- Multilinear in n arguments: For all $1 \leq i \leq n$

$$\begin{aligned} e(h_1, \ldots, h_{i-1}, g_1 g_2, h_{i+1}, \ldots, h_n) \\ = e(h_1, \ldots, h_{i-1}, g_1, h_{i+1}, \ldots, h_n) \\ \cdot e(h_1, \ldots, h_{i-1}, g_2, h_{i+1}, \ldots, h_n) \end{aligned}$$

- Non-degenerate: For all $1 \leq i \leq n$ and all $h_1 \in G_1 \backslash \{1\}$, $\ldots$, $h_n \in G_n \backslash \{1\}$ there is $g \in G_i$ such that

$$e(h_1, \ldots, h_{i-1}, g, h_{i+1}, \ldots, h_n) \neq 1.$$

Some Loose Aspects

- By fixing k arguments we obtain multilinear maps on the remaining $n-k$ arguments, in particular bilinear maps.
- Suppose $G_i \cong \mathbb{Z}/n\mathbb{Z}$ and $G_T \cong \mathbb{Z}/n\mathbb{Z}$. Then a multilinear map takes the form

$$(h_1, \ldots, h_n) \mapsto c h_1 \cdots h_n$$

  for some fixed $c \in \mathbb{Z}/n\mathbb{Z}$.
- So is again essentially ring multiplication of n arguments.

Some Hardness Assumptions

- No efficiently computable isomorphism $G_T \to G_i$.
- Diffie-Hellman variants for $G_1 = \cdots = G_n$: Given

  $$g, g^{a_1}, \ldots, g^{a_{n+1}}$$

  compute $e(g, \ldots, g)^{a_1 \cdots a_{n+1}}$. Or, given

  $$g, g^a$$

  compute $e(g, \ldots, g)^{1/a}$.
- Roughly speaking, any power of g where the exponent cannot be computed as a sum of products of n exponents of input elements should be hard to compute.
- Asymmetric multilinear DDH, SXDH etc.

Point of View

The above discussion can lead to following point of view:

- Use multiplication in $\mathbb{Z}/n\mathbb{Z}$ to get a multilinear map

$$\mathbb{Z}/n\mathbb{Z} \times \cdots \times \mathbb{Z}/n\mathbb{Z} \to \mathbb{Z}/n\mathbb{Z}.$$

- Use an encoding $\mathbb{Z}/n\mathbb{Z} \to G_i$, $a \mapsto g_i^a$ for obfuscation, e.g. $G_i \subseteq E(\mathbb{F}_q)$ and $G_T \subseteq \mathbb{F}_q^\times$.
- But do it in a way such that the group law and the multilinear map can be computed efficiently on the encodings.

This point of view has been taken previously:

- For example, if the Computational Diffie-Hellman problem can be solved in a prime order group, then the prime order group is a "black-box field".
- Was used for a reduction of the DLP to the CDH.

Use homomorphic encryption?

Natural idea:

- Take somewhat homomorphic encryption E on $\mathbb{Z}/n\mathbb{Z}$.
- Encodings are $E(a)$ for $a \in \mathbb{Z}/n\mathbb{Z}$.
- Suppose $E(a+b) = E(a) + E(b)$, $E(ab) = E(a)E(b)$.
- Then multilinear map is

$$(a_1, \dots, a_n) \mapsto E(a_1 \cdots a_n).$$

Problems:

- Problem: Equality test for map values? Does not work if E is secure (indistinguishable cipher texts).
- Easy DLP in arguments ...

Yet, methods from homomorphic encryption yield multilinear maps, as we will see now.

Construction Idea of GGH

Basic idea and features:

- Replace $\mathbb{Z}/n\mathbb{Z}$ by a suitable factor ring R/I.
- Provide a setup of encodings of elements of R/I that partially preserve addition in R/I and allow to compute a k-fold multiplication

$$R/I \times \cdots \times R/I \to R/I$$

  on the level of encodings.
- The equivalent of the DLP is the decoding problem.
- During setup a trapdoor for decoding is constructed.
- The encodings are randomised, there are many encodings of the same element in R/I.
- The output values are also randomised, hence need test for equality and a derandomisation (but not decoding!).

Construction Idea by Way of Example

A suitable but failing example is $R = \mathbb{Z}$.

Let g be small, $I = Rg$ and q a large enough prime.

Let $[x]_q$ with $[x]_q \in [-q/2, q/2)$ and $[x]_q \equiv x \bmod q$.

Let $x^{\mathrm{inv}} \in [-q/2, q/2)$ with $[xx^{\mathrm{inv}}]_q = 1$.

Choose $z \in [0, q]$ random.

> Let x be small. Then
>
> $$[xz^i]_q$$
>
> is called an encoding of $x + I \in R/I$ at level i.

$[xz^i]_q$ looks random, but x can be recovered if z known:

Have $x = [[xz^i]_q (z^{\mathrm{inv}})^i]_q$.

Addition and Multiplication

Addition and multiplication:

- Encodings of same level can be added in R, provided size bound is met for $v + w$:

$$[[vz^i]_q + [wz^i]_q]_q = [(v + w)z^i]_q.$$

- Endodings of different levels can be multiplied in R, provided $i + j \leq k$:

$$[[vz^i]_q \cdot [wz^j]_q]_q = [(vw)z^{i+j}]_q.$$

Setup

Some precomputed elements:

- One element at level one: Let $a \in 1 + I$ be small and define $y = [az]_q$.
- Zero elements at level one: Let $b_i \in 0 + I$ be small and define $x_i = [b_i z]_q$.
- Zero-testing element at level k: Let h somewhat small and coprime to g and define $p_{zt} = [h(z^{\mathrm{inv}})^k g^{\mathrm{inv}}]_q$.

Public versus private:

- The one element y, neutral elements x_i and zero-testing element p_{zt} are made public.
- z, a, b_i, e are kept secret.
- The assertion is that keeping secret works.
- This fails in our example, but we indicate later how GGH solve this.

Sampling and Randomised Encoding

Sampling and randomised encoding at level one:

- Choose small $u \in R$. This represents $u + I$ and is encoding at level 0.
- Multiply with one element and add linear combination of the zeros elements:

$$v = [uy + \sum \lambda_{i,j} x_i]_q$$

  for small random $\lambda_{i,j}$.
- Then v is a randomised encoding of $u + I$ at level one.
- Division by y gives $[u + \sum \lambda_{i,j} b_i a^{\mathrm{inv}}]_q$. Since a^{inv} is big, u cannot be recovered.
- So decoding supposedly hard.

Multilinear Maps

Multilinear map computation:

- The multilinear map takes k randomised encodings at level one and returns their product:

$$(v_1, \ldots, v_n) \mapsto \left[\prod v_i\right]_q .$$

Equality Testing

Zero-testing at level k:

- Equality testing can be reduced to zero testing via subtraction.
- If v encoding at level k then compute

$$[vp_{zt}]_q.$$

  Then this is somewhat small iff v represents $0 + I$.
- Have $[vp_{zt}]_q = [(uz^k)(h(z^{\mathrm{inv}})^k g^{\mathrm{inv}})]_q = [(uh)g^{\mathrm{inv}}]_q$, and this is somewhat small iff $g|u$, hence $u \in I$.
- The use of somewhat small is to ensure that the user cannot produce zero-testing elements at other levels.
- In particular, the product of two somewhat small elements should not be somewhat small anymore.

Extraction

Extraction (Derandomisation):

- Have $[(u-v)p_{zt}]_q$ somewhat small iff u and v encode the same element.
- This means that the most significant bits of $[up_{zt}]_q$ and $[vp_{zt}]_q$ agree.
- These bits can be taken, after the application of a hash function, as unique representing bitstring.

Security

Now $R = \mathbb{Z}$ insecure.

- For example, try all small a to find z.
- g is known from a description of I, thus

$$[p_{zt} g]_q = [h (z^{\text{inv}})^k]_q$$

  This essentially enables distinguishability of level k encodings.
- Small multiples of h^{inv} lead to zero testing parameters at higher level, thus multilinear maps with more arguments.

Idea: Replace R by ring that is $\mathbb{Z}$-module of high rank.

Rings and Ideals in GGH

System data:

- $R = \mathbb{Z}[x]/(x^n + 1)$ with n a power of 2,
- The size $||f||$ of $f \in R$ is the euclidean norm of its vector of coefficients.
- q stays a suitable prime.
- $[f]_q$ the element of R obtained by reducing the coefficients of f modulo q into the interval $[-q/2, q/2)$.
- $I = Rg$ for some small g.

Security of GGH

Security:

- Now exponentially many small a, cannot try all to find z with $y = [az]_q$.
- Small multiples of g and h^{inv} protected by principal ideal problem.
- n needs to be chosen large enough.
- q needs to be chosen large enough to "not interfere with signal".
- q must not be too large to make lattice problems easy (e.g. "lattice gaps").
- Roughly $n = \tilde{O}(k\lambda^2)$ and $q = 2^{n/\lambda}$ for security parameter λ. Thus $q = 2^{c\sqrt{nk}}$.

Properties of GGH

Some rather cool properties:

- Great flexibility in encodings, the set of levels can be any additively closed subset I of $(\mathbb{Z}^{\geq 0})^{\tau}$.
- Zero-testing parameters are provided for a subset of I.

Some restrictive or unfamiliar properties:

- Can assume that HNF-bases of various principal ideals with small generators are known, in particular I. So R/I is known.
- Cannot encode prescribed elements from R/I.
- Incidentally, abelian group DLP in encodings of level zero (the arguments) is easy, since $\cong R/I$.
- Length of encoding not independent of k, no compactness.
- There exists trapdoor for decoding and equality testing.

Outlook

Future developments:

- Will surely see *many* applications.
- Scheme seems efficient. But how efficient can it be?
- Can more efficient techniques for homomorphic encryption for multilinear pairings be used too?

Thank you!