More Efficient Cryptographic Multilinear Maps from Ideal Lattices

Ron Steinfeld
Clayton School of IT
Monash University, Australia
(Based on joint work with A. Langlois and D. Stehlé, ENS Lyon, France)

Monash Discrete Math Group, March 2014

Outline of the talk

1- Introduction
- Background: Cryptographic Multilinear Maps and Applications
- Background: Ideal Lattices

2- Review of GGH construction of approx. multilinear maps

3- GGHLite: Our more efficient construction
- Main ingredients
- Construction
- Asymptotic efficiency
- Using GGHLite in applications

4- Concluding Remarks

Background: Cryptographic Multilinear Maps

Non-interactive Key Exchange (NIKE):

- Alice and Bob want to communicate privately over public channel
- Marvin can see everything sent over the public channel
- Non-interactive setup

Solution: Diffie-Hellman Key Exchange (1976)

- Publish a cyclic group G (generator g, order q) where Discrete Log (DL) problem is hard.
- Alice chooses random $x_1 \in \mathbb{Z}_q$, publishes $y_1 = g^{x_1}$.
- Bob chooses random $x_2 \in \mathbb{Z}_q$, publishes $y_2 = g^{x_2}$.
- Correctness: Both Alice and Bob compute agreed secret key $K = g^{x_1 x_2} = y_1^{x_2} = y_2^{x_1}$.
- Security: Eavesdropper Marvin has to solve the **Computational Diffie-Hellman** problem (CDH),

  CDH: Given g, g^{x_1}, g^{x_2}, compute $g^{x_1 x_2}$.

Background: Cryptographic Multilinear Maps

Non-interactive Key Exchange (NIKE):

- Alice and Bob want to communicate privately over public channel
- Marvin can see everything sent over the public channel
- Non-interactive setup

Solution: Diffie-Hellman Key Exchange (1976)

- Publish a cyclic group G (generator g, order q) where Discrete Log (DL) problem is hard.
- Alice chooses random $x_1 \in \mathbb{Z}_q$, publishes $y_1 = g^{x_1}$.
- Bob chooses random $x_2 \in \mathbb{Z}_q$, publishes $y_2 = g^{x_2}$.
- Correctness: Both Alice and Bob compute agreed secret key $K = g^{x_1 x_2} = y_1^{x_2} = y_2^{x_1}$.
- Security: Eavesdropper Marvin has to solve the **Computational Diffie-Hellman** problem (CDH),

CDH: Given g, g^{x_1}, g^{x_2}, compute $g^{x_1 x_2}$.

Background: Cryptographic Multilinear Maps

21st Century variant (privacy for Facebook): Group of $N > 2$ parties want to communicate privately via 'cloud'.

Solution[J00,BS02]: Use a group where DL is hard and there is an efficient $(N-1)$-linear map $e : G^{N-1} \rightarrow G_T$:

$$e(g^{x_1}, g^{x_2}, \ldots, g^{x_{N-1}}) = e(g, \ldots, g)^{x_1 \cdots x_{N-1}} \forall x_1, \ldots, x_{N-1} \in \mathbb{Z}_q.$$

N-party Non-Interactive Key Exchange

- Publish cyclic groups G, G_T (generators g, g_T, order q) where Discrete Log (DL) problem is hard, with an efficient $(N-1)$-linear map e.
- For $i = 1, \ldots, N$, party P_i chooses $x_i \in \mathbb{Z}_q$, publishes $y_i = g^{x_i}$.
- Correctness: All parties can compute agreed secret key $K = e(g, \ldots, g)^{x_1 \cdots x_N} = e(y_2, y_3, \ldots, y_N)^{x_1}$.
- Security: Hardness of **Multilinear CDH** problem (MCDH),

  MCDH: Given $g, g^{x_1}, \ldots, g^{x_N}$, compute $e(g, \ldots, g)^{x_1 \cdots x_N}$.

Background: Cryptographic Multilinear Maps

21st Century variant (privacy for Facebook): Group of $N > 2$ parties want to communicate privately via 'cloud'.

Solution[J00,BS02]: Use a group where DL is hard and there is an efficient $(N-1)$-linear map $e : G^{N-1} \rightarrow G_T$:

$$e(g^{x_1}, g^{x_2}, \ldots, g^{x_{N-1}}) = e(g, \ldots, g)^{x_1 \cdots x_{N-1}} \forall x_1, \ldots, x_{N-1} \in \mathbb{Z}_q.$$

N-party Non-Interactive Key Exchange

- Publish cyclic groups G, G_T (generators g, g_T, order q) where Discrete Log (DL) problem is hard, with an efficient $(N-1)$-linear map e.
- For $i = 1, \ldots, N$, party P_i chooses $x_i \in \mathbb{Z}_q$, publishes $y_i = g^{x_i}$.
- Correctness: All parties can compute agreed secret key $K = e(g, \ldots, g)^{x_1 \cdots x_N} = e(y_2, y_3, \ldots, y_N)^{x_1}$.
- Security: Hardness of **Multilinear CDH** problem (MCDH),

  MCDH: Given $g, g^{x_1}, \ldots, g^{x_N}$, compute $e(g, \ldots, g)^{x_1 \cdots x_N}$.

Background: Cryptographic Multilinear Maps

21st Century variant (privacy for Facebook): Group of $N > 2$ parties want to communicate privately via 'cloud'.
Solution[J00,BS02]: Use a group where DL is hard and there is an efficient $(N-1)$-linear map $e : G^{N-1} \to G_T$:

$$e(g^{x_1}, g^{x_2}, \ldots, g^{x_{N-1}}) = e(g, \ldots, g)^{x_1 \cdots x_{N-1}} \forall x_1, \ldots, x_{N-1} \in \mathbb{Z}_q.$$

N-party Non-Interactive Key Exchange

- Publish cyclic groups G, G_T (generators g, g_T, order q) where Discrete Log (DL) problem is hard, with an efficient $(N-1)$-linear map e.
- For $i = 1, \ldots, N$, party P_i chooses $x_i \in \mathbb{Z}_q$, publishes $y_i = g^{x_i}$.
- Correctness: All parties can compute agreed secret key $K = e(g, \ldots, g)^{x_1 \cdots x_N} = e(y_2, y_3, \ldots, y_N)^{x_1}$.
- Security: Hardness of **Multilinear CDH** problem (MCDH),
  MCDH: Given $g, g^{x_1}, \ldots, g^{x_N}$, compute $e(g, \ldots, g)^{x_1 \cdots x_N}$.

Background: Cryptographic Multilinear Maps – History

- 2000: Bilinear ($k = 2$) via Weil pairings on algebraic curves, applications:
  - 2000: 3-party non-interactive key agreement [J00]
  - 2000-2001: Identity-Based Encryption (IBE) [SK00,BF01]
  - 2001: Short signatures [BS01]
  - 2000-2013: **lots** of others
- 2002: Applications for k-linear maps [BS02]
  - $(k + 1)$-party non-interactive key agreement
  - Efficient Broadcast Encryption
  - and others...
- 2012: First plausible realization for $k > 2$, via ideal lattices [GGH12], applications:
  - 2012-2013: Functional Encryption for arbitrary functions
  - 2013: Program obfuscation notions for arbitrary functions
- 2014: GGHLite – More efficient variant of GGH construction (this talk)

Approx. Multilin. Maps: GGH 'Graded Encoding Scheme'

GGH realization: not quite a k-linear map, but essentially the same

Technically, a k-graded encoding scheme:

- Replace groups $\mathbb{Z}_q, G$ by
  - Rings R_g, R_q and some public parameters par.
- Replace 'Encode $x \in \mathbb{Z}_q$ as $g^x \in G$' by
  - 'Encode $x \in R_g$ as $\mathsf{Enc}_1(\mathrm{par}, x; \rho) \in R_q$' – randomized 'level 1 encoding' of 'level 0' element x using randomness ρ.
- Replace $e(g_1^{x_1}, \ldots, g_k^{x_k}) = e(g_1, \ldots, g_k)^{x_1 \cdots x_k}$ by
  - Homomorphic up to 'level k':
    $$\mathsf{Enc}_1(\mathrm{par}, x_1; \rho_1) \cdots \mathsf{Enc}_1(\mathrm{par}, x_k; \rho_k) = \mathsf{Enc}_k(\mathrm{par}, x_1 \cdots x_k; \rho)$$
    and
    $$x \cdot \mathsf{Enc}_k(\mathrm{par}, z; \rho) = \mathsf{Enc}_k(\mathrm{par}, x \cdot z; \rho'), \text{ for any } x \in R_g.$$
  - Randomness-independent extraction at level k – $\mathsf{Ext}(\mathrm{par}, \mathsf{Enc}_k(\mathrm{par}, x; \rho)) = r(x) \in \{0,1\}^n$ is independent of randomness ρ, and uniformly random for $x \hookleftarrow U(R_g)$.

Multilinear Maps: GGH 'Graded Encoding Scheme'

N-party NIKE from $N-1$-Graded Encoding Scheme:

- Publish rings R_g, R_q and pub. params. par of $N-1$-Graded Encoding Scheme.
- For $i = 1, \ldots, N$, party P_i chooses $x_i \in R_g$, publishes $y_i = \mathsf{Enc}_1(\mathrm{par}, x_i; \rho_i)$.
- Correctness: All parties can compute agreed secret key

  $$K = \mathsf{Ext}(\mathrm{par}, \mathsf{Enc}_{N-1}(\mathrm{par}, x_1 \cdots x_N; \rho)) = \mathsf{Ext}(\mathrm{par}, x_1 \cdot y_2 \cdot y_3 \cdots y_N)$$

- Security: To compute K, eavesdropper Marvin has to solve the **Extraction Graded Computational Diffie-Hellman** problem – **Ext-GCDH**: Given $\mathrm{par}, y_1 = \mathsf{Enc}_1(\mathrm{par}, x_1; \rho_1), \ldots, y_N = \mathsf{Enc}_1(\mathrm{par}, x_N; \rho_N)$, compute $\mathsf{Ext}(\mathrm{par}, \mathsf{Enc}_{N-1}(\mathrm{par}, x_1 \cdots x_N; \rho))$.

Polynomial Rings

Take $\phi \in \mathbb{Z}[x]$ monic of degree n.

$$R^{\phi} \ := \ \Big[\mathbb{Z}[x]/(\phi), +, \times\Big].$$

Interesting ϕ's:

- $\phi = x^n - 1 \ \rightarrow \ R^{-}, \quad \phi = x^n + 1 \ \rightarrow \ R^{+}.$
- For n a power of 2, the ring R^{+} is isomorphic to the ring of integers of $K = \mathbb{Q}[e^{i\pi/n}]$:

$$K \simeq \mathbb{Q}[x]/(x^n + 1)$$
$$\mathcal{O}_K \simeq \mathbb{Z}[x]/(x^n + 1).$$

$\Rightarrow$ Rich algebraic structure (great for design and proofs).

Polynomial Rings

Take $\phi \in \mathbb{Z}[x]$ monic of degree n.

$$R^{\phi} := \left[\mathbb{Z}[x]/(\phi), +, \times\right].$$

Interesting ϕ's:

- $\phi = x^n - 1 \rightarrow R^{-}, \quad \phi = x^n + 1 \rightarrow R^{+}$.
- For n a power of 2, the ring R^{+} is isomorphic to the ring of integers of $K = \mathbb{Q}[e^{i\pi/n}]$:

$$\begin{aligned} K &\simeq \mathbb{Q}[x]/(x^n + 1) \\ \mathcal{O}_K &\simeq \mathbb{Z}[x]/(x^n + 1). \end{aligned}$$

$\Rightarrow$ Rich algebraic structure (great for design and proofs).

Polynomial Rings

Take $\phi \in \mathbb{Z}[x]$ monic of degree n.

$$R^{\phi} := \left[\mathbb{Z}[x]/(\phi), +, \times\right].$$

Interesting ϕ's:

- $\phi = x^n - 1 \rightarrow R^-, \quad \phi = x^n + 1 \rightarrow R^+.$
- For n a power of 2, the ring R^+ is isomorphic to the ring of integers of $K = \mathbb{Q}[e^{i\pi/n}]$:

$$\begin{aligned} K &\simeq \mathbb{Q}[x]/(x^n + 1) \\ \mathcal{O}_K &\simeq \mathbb{Z}[x]/(x^n + 1). \end{aligned}$$

$\Rightarrow$ Rich algebraic structure (great for design and proofs).

Polynomial Rings

Let $q \geq 2$ and $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$.

$$R_q^{\phi} := \left[\mathbb{Z}_q[x]/(\phi), +, \times\right].$$

- Arithmetic in R_q^{ϕ} costs $\widetilde{O}(n \log q)$.
- R_q^{+} is isomorphic to $\mathcal{O}_K/(q)$.

Polynomial Rings

Let $q \geq 2$ and $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$.

$$R_q^{\phi} \;:=\; \Big[\mathbb{Z}_q[x]/(\phi), +, \times\Big].$$

- Arithmetic in R_q^{ϕ} costs $\widetilde{O}(n \log q)$.
- R_q^{+} is isomorphic to $\mathcal{O}_K/(q)$.

Lattices Background: Approx-SVP

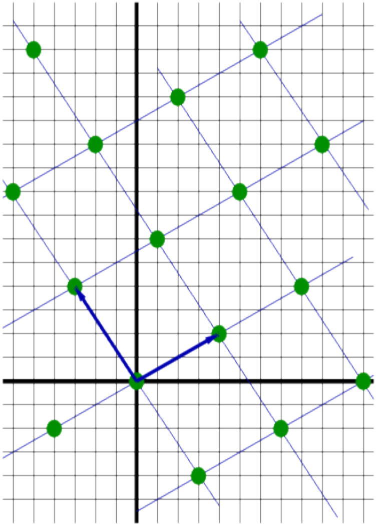
Lattice $\equiv \{\sum_{i \leq n} x_i \mathbf{b}_i : x_i \in \mathbb{Z}\}$,
for some lin. independent $\mathbf{b}_i$'s.

Minimum: $\lambda(L) = \min(\|\mathbf{b}\| : \mathbf{b} \in L \setminus \mathbf{0})$

γ-SVP

Find $\mathbf{b} \in L$ with: $0 < \|\mathbf{b}\| \leq \gamma \cdot \lambda(L)$.

- No known sub-exp. algorithm for $\gamma = Poly(n)$.
- Not even quantumly.
- Seems harder than Int-Fac and DLog.



Lattices Background: Approx-SVP

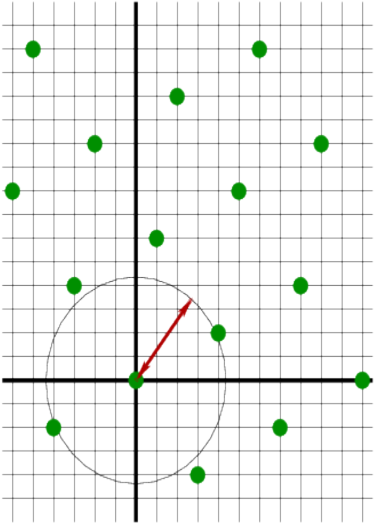
Lattice $\equiv \{\sum_{i \leq n} x_i \mathbf{b}_i : x_i \in \mathbb{Z}\}$,
for some lin. independent $\mathbf{b}_i$'s.

Minimum: $\lambda(L) = \min(\|\mathbf{b}\| : \mathbf{b} \in L \setminus \mathbf{0})$

γ-SVP

Find $\mathbf{b} \in L$ with: $0 < \|\mathbf{b}\| \leq \gamma \cdot \lambda(L)$.

- No known sub-exp. algorithm for $\gamma = \mathcal{P}oly(n)$.
- Not even quantumly.
- Seems harder than Int-Fac and DLog.



Lattices Background: Approx-SVP

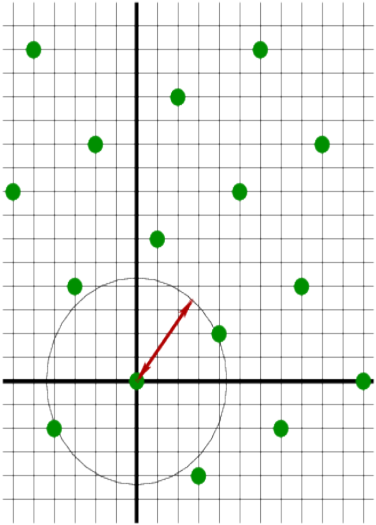
Lattice $\equiv \{\sum_{i \le n} x_i \mathbf{b}_i : x_i \in \mathbb{Z}\}$,
for some lin. independent $\mathbf{b}_i$'s.

Minimum: $\lambda(L) = \min(\|\mathbf{b}\| : \mathbf{b} \in L \setminus \mathbf{0})$

γ-SVP

Find $\mathbf{b} \in L$ with: $0 < \|\mathbf{b}\| \le \gamma \cdot \lambda(L)$.

- No known sub-exp. algorithm for $\gamma = \mathcal{P}oly(n)$.
- Not even quantumly.
- Seems harder than Int-Fac and DLog.



Lattices Background: Approx-Ideal-SVP

- $I \subseteq R^{\phi}$ is an ideal if:

$$\forall a, b \in I, \forall r \in R^{\phi} : \quad a + b \cdot r \in I.$$

- We identify polynomials to vectors via their coefficients:

$$\begin{array}{ccc} R^{\phi} & \rightarrow & \mathbb{Z}^n \\ \sum_{i<n} f_i x^i & \mapsto & (f_0, \ldots, f_{n-1})^t \end{array}$$

- An ideal I can be viewed as a lattice, called an ideal lattice.

$\mathcal{P}oly(n)$-Ideal-SVP: $\mathcal{P}oly(n)$-SVP restricted to ideal lattices.

No significant computational advantage known for this general family of inputs.

Lattices Background: Approx-Ideal-SVP

- $I \subseteq R^{\phi}$ is an ideal if:

$$\forall a, b \in I, \forall r \in R^{\phi}: \quad a + b \cdot r \in I.$$

- We identify polynomials to vectors via their coefficients:

$$\begin{array}{ccc} R^{\phi} & \to & \mathbb{Z}^n \\ \sum_{i<n} f_i x^i & \mapsto & (f_0, \ldots, f_{n-1})^t \end{array}$$

- An ideal I can be viewed as a lattice, called an ideal lattice.

$\mathcal{P}oly(n)$-Ideal-SVP: $\mathcal{P}oly(n)$-SVP restricted to ideal lattices.

No significant computational advantage known for this general family of inputs.

Lattices Background: Approx-Ideal-SVP

- $I \subseteq R^{\phi}$ is an ideal if:

$$\forall a, b \in I, \forall r \in R^{\phi} : \quad a + b \cdot r \in I.$$

- We identify polynomials to vectors via their coefficients:

$$\begin{array}{ccc} R^{\phi} & \rightarrow & \mathbb{Z}^n \\ \sum_{i<n} f_i x^i & \mapsto & (f_0, \ldots, f_{n-1})^t \end{array}$$

- An ideal I can be viewed as a lattice, called an ideal lattice.

$\mathcal{P}oly(n)$-Ideal-SVP: $\mathcal{P}oly(n)$-SVP restricted to ideal lattices.

No significant computational advantage known for this general family of inputs.

Lattices Background: Approx-Ideal-SVP

- $I \subseteq R^{\phi}$ is an ideal if:

$$\forall a, b \in I, \forall r \in R^{\phi} : \quad a + b \cdot r \in I.$$

- We identify polynomials to vectors via their coefficients:

$$\begin{array}{ccc} R^{\phi} & \rightarrow & \mathbb{Z}^n \\ \sum_{i<n} f_i x^i & \mapsto & (f_0, \ldots, f_{n-1})^t \end{array}$$

- An ideal I can be viewed as a lattice, called an ideal lattice.

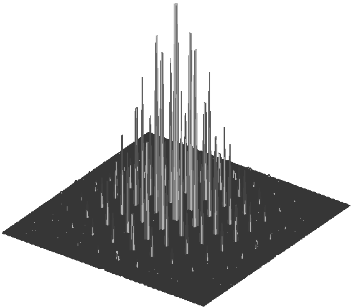
$\mathcal{P}oly(n)$-Ideal-SVP: $\mathcal{P}oly(n)$-SVP restricted to ideal lattices.

No significant computational advantage known for this general family of inputs.

Lattices Background: Discrete Gaussian Distributions

$D_{L,S,c}$ denotes discrete Gaussian distrib. on n-dim. lattice L, full-rank deviation matrix $S \in \mathbb{R}^{n \times n}$, centre c (sample using [GePeVa'08]):

$$\forall x \in L: \quad D_{L,S,c}[x] \sim \exp\left(-\pi(x-c)^T(S^T S)^{-1}(x-c)\right).$$



Approx. Multilin. Maps: GGH k-graded encoded scheme

Public Parameters Generation:

- Sample 'small' $g \hookleftarrow D_{R,\sigma}$ until $\|g^{-1}\| \leq \ell_{g^{-1}}$ and $I = \langle g \rangle$ is a prime ideal. Define encoding domain $R_g = R/\langle g \rangle$.
- Sample $z \hookleftarrow U(R_q)$.
- Sample a level-1 encoding of 1: set $y = [a \cdot z^{-1}]_q$ with $a \hookleftarrow D_{1+I,\sigma'}$.
- Sample m_r level-1 encodings of 0: set $x_j = [b_j \cdot z^{-1}]_q$ with $b_j \hookleftarrow D_{I,\sigma'}$ for all $j \leq m_r$.
- Sample $h \hookleftarrow D_{R,\sqrt{q}}$ and define the zero-testing parameter $p_{zt} = [\frac{h}{g} z^k]_q \in R_q$.
- Return $\mathrm{par} = (n, q, y, \{x_j\}_{j \leq m_r})$ and p_{zt}.

Approx. Multilin. Maps: GGH k-graded encoded scheme

Level-1 encoding $\mathrm{Enc}_1(\mathrm{par}, e)$: Given level-0 $e \in R$:

- Encode e at level 1: $u' = [e \cdot y]_q$ (note $u' = [c'/z]_q$ with $c' \in e + I$).
- Re-randomize: Sample small $\rho_j \hookleftarrow D_{\mathbb{Z},\sigma_1^*}$ for $j \leq m_r$ and return $u = [u' + \sum_{j=1}^{m_r} \rho_j x_j]_q$.
  (Note $u = [c/z]_q$ with $c \in e + I$ and $c = c' + \sum_j \rho_j b_j$.)

Multiplying encodings mult: Given level-k_1 encoding $u_1 = [c_1/z^{k_1}]_q$ and level-k_2 encoding $u_2 = [c_2/z^{k_2}]_q$:

- Return $u = [u_1 \cdot u_2]_q$, a level-$(k_1 + k_2)$ encoding of $[c_1 \cdot c_2]_g$.
  (note $u_1 \cdot u_2 = [c_1 c_2/z^{k_1+k_2}]_q$ and $c_1 \cdot c_2 \in e_1 \cdot e_2 + I$).

Approx. Multilin. Maps: GGH k-graded encoded scheme

Extraction at level k $\mathrm{Ext}(\mathrm{par}, u)$: Given a level-$k$ encoding $u = [c/z^k]_q$, return $v = \mathsf{MSB}_\ell([p_{zt} \cdot u]_q)$ with $\ell < (1/4 - \varepsilon) \log q$.

Correctness of extraction:

- At level 1: if $c = [c]_g + gr$ for some **small** $r \in R$, then $v = \mathsf{MSB}_\ell(\frac{h}{g}([c]_g + gr)) = \mathsf{MSB}_\ell(\frac{h}{g}[c]_g + hr)$, which is equal to $\mathsf{MSB}_\ell(\frac{h}{g}[c]_g)$, with high probability if $q > \|r\|^8$.
- After k multiplications:
  - Let $u_i = [\frac{x_i + g \cdot r_i}{z}]_q$ for $i = 1, \ldots, k$ be encodings of $x_1, \ldots, x_k$.
  - For $u \stackrel{\text{def}}{=} u_1 \cdot u_2 \cdots u_k = [\frac{x + g \cdot r}{z^k}]_q$ to be a valid encoding of $x = x_1 \cdots x_k$, need $\|r\|$ to stay **small** compared to q:

    $$\|r\| = O(2^k \cdot \|(g \cdot r_1) \cdots (g \cdot r_k)\|) = O((\mathcal{P}oly(n) \cdot N)^k) < q^{1/8}.$$

    where $N \stackrel{\text{def}}{=} \max_i \|g \cdot r_i\|$.

Approx. Multilin. Maps: GGH k-graded encoded scheme

Security of GDH for GGH scheme: not well understood.
Known attack needs 'small' multiple d of g ($\|d \cdot g\| < q$).

- **Fact:** Easy [GGH12] to compute basis for $\langle g \rangle$ from par .
- **Conclusion:** Security relies on hardness of q-ideal-SVP.

Attack on 'Graded Discrete Log' prob. given
$u = \mathsf{Enc}_1(\mathrm{par}, x; r) = [\frac{x+r\cdot g}{z}]_q$ (idea):

- Compute $p'_{zt} \stackrel{\mathrm{def}}{=} [d \cdot g \cdot p_{zt}]_q = [(d \cdot g) \cdot (\frac{h}{g} z^k)]_q = [d \cdot h \cdot z^k]_q$.
- Lift: $u' = [u \cdot y^{k-1}]_q = [\frac{x+r' \cdot g}{z^k}]_q$, $y' = [y^k]_q = [\frac{1+r'_y \cdot g}{z^k}]_q$.
- Compute $u'' = [u' \cdot p'_{zt}]_q = d \cdot h \cdot (x + r' \cdot g) \in R$ and $y'' = [y' \cdot p'_{zt}]_q = d \cdot h \cdot (1 + r'_y \cdot g) \in R$.
- Using basis for $\langle g \rangle$, easy to compute a ('large') rep. $x' \in R$ with $x' \equiv u'' \cdot (y'')^{-1} \bmod \langle g \rangle$, so $x' \equiv x \bmod \langle g \rangle$.
- Compute a 'small' rep. $x'' = x' \bmod \langle d \cdot g \rangle$ with $x'' \equiv x \bmod \langle g \rangle$.

GGHLite: Main Ingredients

We improve encoding re-randomization in GGH:

- Pub. Pars. contain level-1 encodings of 0, namely $\{x_j = [b_j/z]_q\}_{j \leq m_r}$ and level-1 encoding of 1, namely y.
- To randomize level-1 encoding $u' = [e \cdot y]_q$, output $u = [u' + \sum_j \rho_j x_j]_q = [c/z]_q$ with $c = c' + \sum_j \rho_j b_j$.
- Randomizers ρ_j's are sampled from a discrete Gaussian distribution over $\mathbb{Z}$ with deviation parameter σ^*.

Re-randomization is essential for security of GDH:

- Without re-randomization, e can be be efficiently recovered from $u' = [e \cdot y]_q$ and y ($u = [u' y^{-1}]_q$).
- Re-randomization can prevent this attack.

GGHLite: First Main Ingredient

But, how to choose the re-randomization parameters for security level 2^λ?

Question: How large should re-randomization deviation σ^* be?

- in GGH, exponential drowning: $\sigma^*/\|c'\| \geq 2^\lambda$
- Makes distribution of u (almost) independent of u'
- But incurs severe efficiency penalty.
  - Need $q \geq 2^\lambda$.
  - Security of q-ideal-SVP deteriorates exponentially with $\log q$.
  - Need quadratic dimension: $n \geq \lambda^2$!

GGHLite First Ingredient: We show that polynomial drowning is sufficient for security: $\sigma^*/\|c'\| \geq \mathcal{P}oly(\lambda)$

But, our analysis only seems to apply to computational GDH problem.

- We use Rényi Divergence in place of Statistical Distance in analysing re-randomized distribution vs. 'canonical' one

GGHLite: Second Main Ingredient

Question: How many encodings of 0 are needed?

GGH construction:

- Needs $m_r = \Omega(n \log n)$ encodings of 0
- Uses rational integer Gaussian randomizers ($\rho_j \in \mathbb{Z}$) as coefficients
- Uses a 'discrete Gaussian Leftover Hash Lemma' to show $\sum_{j \le m_r} \rho_j b_j$ distrib. is close to a discrete Gaussian on I

GGHLite Second Ingredient: $m_r = 2$ encodings of 0 are sufficient

- Uses Gaussian randomizers over full ring ($\rho_j \in R$)
- New algebraic variant of 'discrete Gaussian Leftover Hash Lemma' over R: we show $\sum_{j \le m_r} \rho_j b_j$ distribution is close to a discrete Gaussian on I

GGHLite: Our simplified k-graded encoded scheme

Public Parameters Generation:

- Sample $g \hookleftarrow D_{R,\sigma}$ until $\|g^{-1}\| \leq \ell_{g^{-1}}$ and $I = \langle g \rangle$ is prime.
- Sample $z \hookleftarrow U(R_q)$.
- Sample a level-1 encoding of 1: $y = [a \cdot z^{-1}]_q$ with $a \hookleftarrow D_{1+I,\sigma'}$.
- Sample $B = (b_1, b_2)$ from $(D_{I,\sigma'})^2$. If $\langle b_1, b_2 \rangle \neq I$, or $\sigma_n(\mathrm{rot} B) < \ell_b$, then re-sample.
- Define level-1 encodings of 0: $x_1 = [b_1 \cdot z^{-1}]_q$, $x_2 = [b_2 \cdot z^{-1}]_q$.
- Sample $h \hookleftarrow D_{R,\sqrt{q}}$ and define the zero-testing parameter $p_{zt} = [\frac{h}{g} z^k]_q \in R_q$.
- Return $\mathrm{par} = (n, q, y, x_1, x_2, p_{zt})$.

Level-1 encoding $\mathrm{Enc}_1(\mathrm{par}, e)$: Given level-0 $e \in R$:

- Encode e at level 1: Compute $u' = [e \cdot y]_q$.
- Return $u = [(u' + \rho_1 \cdot x_1 + \rho_2 \cdot x_2)/z]_q$, with $\rho_1, \rho_2 \hookleftarrow D_{R,\sigma_1^*}$.

GGHLite: Formalizing Re-randomization Security

How to formalize re-randomization security requirement?

Informal req.: Prevent correlation of statistical properties of re-randomized encoding with encoded element.

Formal req.: Breaking Ext-GCDH problem is as hard as breaking canonical Ext-GCDH problem

- **Ext-GCDH**: Given $\mathrm{par}, y_1 = [e_1 \cdot y + \rho_{1,1} \cdot x_1 + \rho_{2,1} \cdot x_2]_q, \ldots, y_N = [e_N \cdot y + \rho_{1,N} \cdot x_1 + \rho_{2,N} \cdot x_2]_q$, compute $\mathsf{Ext}(\mathrm{par}, \mathsf{Enc}_{N-1}(\mathrm{par}, x_1 \cdots x_N; \rho)) = MSB_\ell(p_{zt} \cdot e_1 \cdots e_N)$.
- canonical **Ext-GCDH**: Given $\mathrm{par}, y_1 = [c_1 z^{-1}]_q, \ldots, y_N = [c_N z^{-1}]_q$ with $c_i \hookleftarrow D_{I + e_i, \sigma_1^* B^T}$ for $i = 1, \ldots, N$, compute $\mathsf{Ext}(\mathrm{par}, \mathsf{Enc}_{N-1}(\mathrm{par}, x_1 \cdots x_N; \rho)) = MSB_\ell(p_{zt} \cdot e_1 \cdots e_N)$.

Theorem. This requirement is satisfied, i.e. such a reduction exists for GGHLite, under suitable parameter conditions.

GGHLite Re-randomization Security: First Ingredient

D_1: distrib. of $y_i = [v_i/z]_q$ in **Ext-GCDH** problem

- v_i distrib. $\approx D_{I+e_i,\sigma_1^* B^T, c_i'}$ – 'small' centre c_i'.

D_2: distrib. of $y_i = [v_i/z]_q$ in canonical **Ext-GCDH** problem

- v_i distrib. $\approx D_{I+e_i,\sigma_1^* B^T}$ – zero centre.

GGH strong requirement based on statistical distance (SD) Δ:

$$\Delta(D_1, D_2) \stackrel{\text{def}}{=} \sum_x |D_1(x) - D_2(x)| \le 2^{-\lambda},$$

Prob. Preservation Property of SD: Any adversary A with succ. prob. ε against **Ext-GCDH** problem, has succ. prob. ε' against canonical **Ext-GCDH** problem with:

$$\varepsilon' \ge \varepsilon - \Delta(D_1, D_2) \ge \varepsilon - 2^{-\lambda},$$

- To handle $\varepsilon = 2^{-\lambda}$, need $\Delta(D_1, D_2) < 2^{-\lambda}$!
- Consequently, need $\frac{\sigma_1^*}{\|c_i'\|} = 2^{\Omega(\lambda)}$ (exponential drowning).

GGHLite Re-randomization Security: First Ingredient

D_1: distrib. of $y_i = [v_i/z]_q$ in **Ext-GCDH** problem

- v_i distrib. $\approx D_{I+e_i,\sigma_1^* B^T,c_i'}$ – 'small' centre c_i'.

D_2: distrib. of $y_i = [v_i/z]_q$ in canonical **Ext-GCDH** problem

- v_i distrib. $\approx D_{I+e_i,\sigma_1^* B^T}$ – zero centre.

GGHLite weak requirement based on Rényi divergence (RD) R:

$$R(D_1 \| D_2) \stackrel{\text{def}}{=} \sum_x D_1^2(x)/D_2(x) \le \mathcal{P}oly(\lambda),$$

Prob. Preservation Property of RD: Any adversary A with succ. prob. ε against **Ext-GCDH** problem, has succ. prob. ε' against canonical **Ext-GCDH** problem with:

$$\varepsilon' \ge \varepsilon / R(D_1 \| D_2)^2 \ge \varepsilon / \mathcal{P}oly(\lambda),$$

- Useful even if $\varepsilon < R(D_1, D_2)^{-1}$ – use $R(D_1 \| D_2) \le \mathcal{P}oly(\lambda)$.
- We show: $R(D_1 \| D_2) \le \exp\left(2\pi \|c_i'\|^2 / \sigma_n(\sigma_1^* B^T)^2\right)$.
- For $R(D_1 \| D_2) \le \mathcal{P}oly(\lambda)$, can use $\frac{\sigma_1^*}{\|c_i'\|} = O(\frac{1}{\log \lambda})$.

GGHLite Re-randomization Security: Second Ingredient

D_1: distrib. of $y_i = [v_i/z]_q$ in **Ext-GCDH** problem

- v_i distrib. $\approx D_{I+e_i,\sigma_1^* B^T, c_i'}$ – 'small' centre c_i'.

In actual scheme $(e_i \cdot a + \rho_1 \cdot b_1 + \rho_2 \cdot b_2)/z]_q$ with $\rho_i \sim D_{R,\sigma_1^*}$.
How do we show $\rho_1 \cdot b_1 + \rho_2 \cdot b_2 \approx D_{I,\sigma_1^* B^T}$ $(B = g \cdot [t_1, t_2] \in R^2)$?

- **Step 1:** Show $T \cdot R^2 = [t_1, t_2] \cdot R^2 = R$, except for some constant probability < 1.
  - Probability that two 'random' algebraic integers are co-prime ($\approx \zeta_R(2)^{-1}$).
- **Step 2:** Study the 'orthogonal' lattice $A_T = \{v \in R^2 : T \cdot v = 0\}$.
  - Use equality of Minkowski minima of A_T to bound 'smoothing parameter' $\eta_\varepsilon(A_T)$.
  - Apply known results [AGHS12] on 'smoothing of Gaussians modulo a lattice': If $\sigma_1^* > \eta_\varepsilon(A_T)$, then $\rho_1 \cdot t_1 + \rho_2 \cdot t_2$ is within SD 2ε of $D_{R,\sigma_1^* T^T}$.

GGHLite: Asymptotic Parameters

Parameter	GGHLite	GGH
m_r	2	$\Omega(n \log n)$
σ	$O(n \log n)$	$O(n \log n)$
$\ell_{g^{-1}}$	$O(1/\sqrt{n \log n})$	$O(1/\sqrt{n \log n})$
$\varepsilon_d, \varepsilon_e, \varepsilon_\rho$	$O(k^{-1})$	$O(2^{-\lambda} k^{-1})$
σ'	$\widetilde{O}(n^{2.5})$	$\widetilde{O}(n^{1.5}\sqrt{\lambda})$
σ_1^*	$\widetilde{O}(n^{4.5}\sqrt{\log k})$	$\widetilde{O}(2^\lambda n^{4.5}(\lambda + \log k))$
ε_{ext}	$O(\lambda^{-\omega(1)})$	$O(\lambda^{-\omega(1)})$
q	$\widetilde{O}((n^{8.5}\sqrt{\log k})^{8k})$	$\widetilde{O}((2^\lambda n^8 \lambda^{1.5})^{8k})$
n	$O(k\lambda \log \lambda)$	$O(k\lambda^2)$
$\lvert\text{enc}\rvert$	$O(k^2 \lambda \log^2(k\lambda))$	$O(k^2\lambda^3)$
$\lvert\text{par}\rvert$	$O(k^3 \lambda \log^2(k\lambda))$	$O(k^3\lambda^5 \log(k\lambda))$

Adapting Applications of GGH to GGHLite

Applications often need semantic security: no **partial** information on key leaks.

GGH security analysis applies to Graded Decision Diffie-Hellman problem (GDDH): Distinguish between the distributions

$$\mathcal{D}_{DDH} = \{\mathrm{par}, (u_i = \mathsf{Enc}_1(x_i))_{0 \leq i \leq k}, v = \mathsf{Enc}_1(x_0 \cdot x_1 \cdots x_k)\}$$

and

$$\mathcal{D}_R = \{\mathrm{par}, (u_i = \mathsf{Enc}_1(x_i))_{0 \leq i \leq k}, v = \mathsf{Enc}_1(f_0)\}$$ for indep. unif. dist. f_0.

GGHLite security analysis only applies to Extraction Graded Computational Diffie-Hellman problem (Ext-GCDH).

Adapting Applications of GGH to GGHLite

Question: How to adapt GGH app. to rely on Ext-GCDH rather than GDDH?

Answer: Replace agreed key $K = \mathsf{Ext}(\mathrm{par}, v)$ in original protocol by

$$K = H(\mathsf{Ext}(\mathrm{par}, v))$$

in modified protocol, where $H(\cdot)$ is a cryptographic hash function. If $H(\cdot)$ is modelled as a black-box random function ('Random Oracle Model'), then security of modified protocol relies on Ext-GCDH – our GGHLite analysis applies!

Conclusions

Presented GGHLite, a more efficient variant of GGH graded encoding scheme.

Open Problems:

- Can our Rényi divergence analysis be applied to the Decision Graded Diffie Hellman problem?
- Understand the complexity of our canonical Ext-GCDH problem – provable relation to well studied lattice problems?
- Alternative constructions for graded encoding scheme, with provable security from standard lattice problems?
- Understand relation beteen GGH/GGHLite and more recent 'Jigsaw puzzle' variants (obfuscation).
- Concrete computational / space efficiency of GGHLite based on best known attacks?