GGHLite: More Efficient Multilinear Maps from Ideal Lattices

Adeline Langlois[1] Damien Stehlé[1] Ron Steinfeld[2]

[1]LIP, ENS de Lyon, France

[2]Monash University, Australia

May 13, 2014

Our main result

Decrease size of public parameters from $O(\lambda^5 \log \lambda)$ to $O(\lambda \log^2 \lambda)$

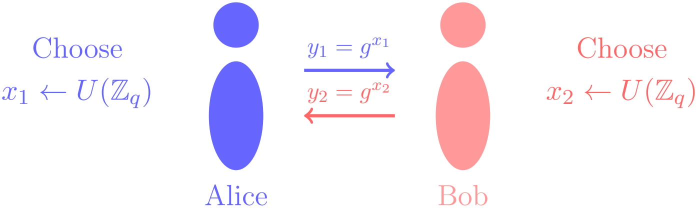
Lower size of parameters and finer security analysis

A more efficient cryptographic multilinear maps, obtained by formalizing, simplifying and improving the re-randomization process in the GGH construction.

For each encoding

Garg, Gentry and Halevi 2013

Diffie-Hellman Key Exchange (1976)

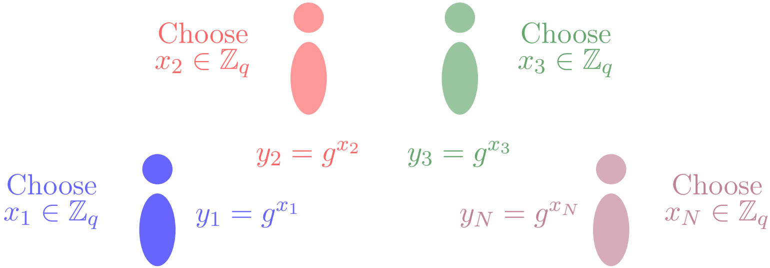


Agreed secret key: $K = g^{x_1 x_2} = y_1^{x_2} = y_2^{x_1}$

- Security: **Decisional Diffie-Hellman** problem,

  DDH: For $x_1, x_2, x_3 \leftarrow U(\mathbb{Z}_q)$, distinguish between $(g^{x_1}, g^{x_2}, g^{x_1 x_2})$ and $(g^{x_1}, g^{x_2}, g^{x_3})$.

Cryptographic Multilinear Maps – 21st Century variant



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 Decisional DH** problem,

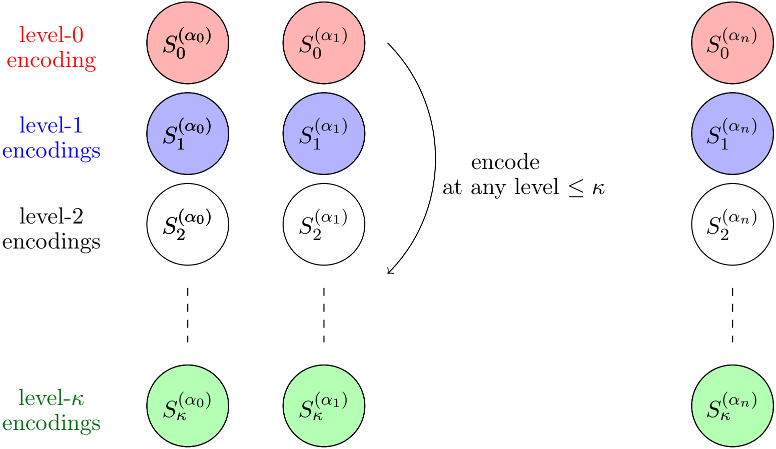
  MDDH: For $x_1, \ldots, x_N, x' \leftarrow U(\mathbb{Z}_q)$, distinguish between $(g^{x_1}, \ldots, g^{x_N}, e(g, \ldots, g)^{x_1 \cdots x_N})$ and $(g^{x_1}, \ldots, g^{x_N}, e(g, \ldots, g)^{x'})$.

Cryptographic Multilinear Maps – History

- 2000: Applications for elliptic curves pairings ($\kappa = 2$)
  - 2000: 3-party non-interactive key agreement [Joux00],
  - 2000-2001: Identity-Based Encryption (IBE) ... [SakaiOhgishiKasahara00,BonehFranklin01],
- 2002: Applications for κ-linear maps [BonehSilverberg03]
  - $(\kappa + 1)$-party non-interactive key agreement ...
- 2012: [GargGentryHalevi13]
  - First plausible realization for $\kappa > 2$, via ideal lattices,
  - Applications:
    - 2012-2013: Functional Encryption for arbitrary functions,
    - 2013: Program obfuscation notions for arbitrary functions.
- 2013: Variant over the integers [CoronLepointTibouchi13].
- 2014: GGHLite – More efficient variant of GGH (this talk).

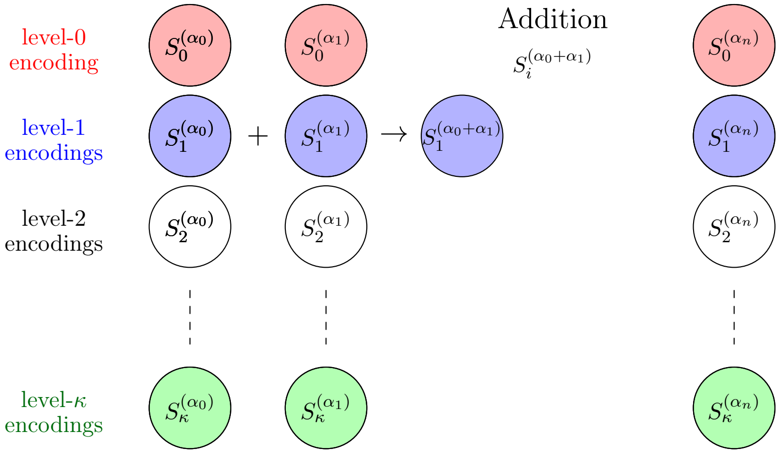
κ-Graded encoding scheme

A ring R_{Plain} of plaintext and a ring R_{Enc} of encodings, with a system of sets $\mathcal{S} = \{S_i^{(\alpha)} \subseteq R_{\text{Enc}} : \alpha \in R_{\text{Plain}}, 0 \leq i \leq \kappa\}$.



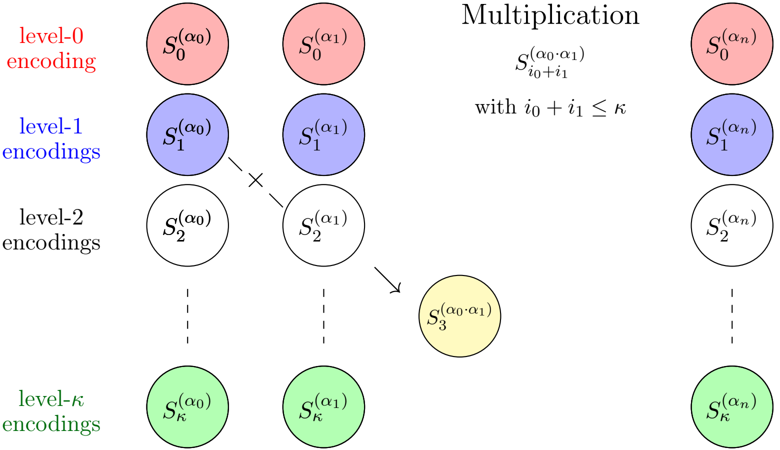
κ-Graded encoding scheme

A ring R_{Plain} of plaintext and a ring R_{Enc} of encodings, with a system of sets $\mathcal{S} = \{S_i^{(\alpha)} \subseteq R_{\text{Enc}} : \alpha \in R_{\text{Plain}}, 0 \leq i \leq \kappa\}$.



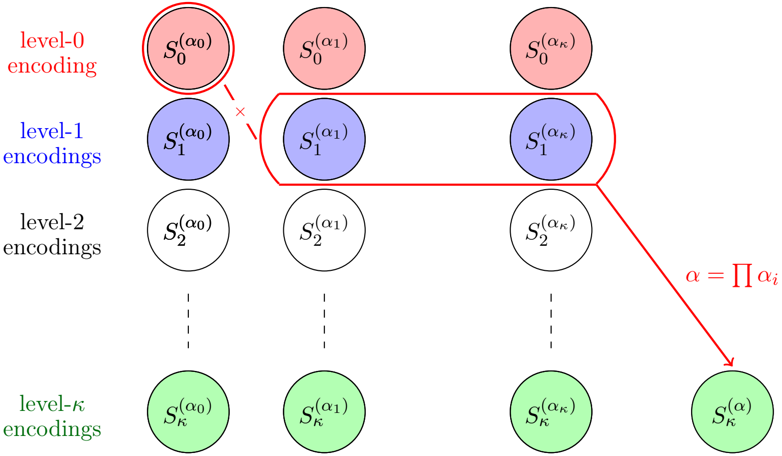
κ-Graded encoding scheme

A ring R_{Plain} of plaintext and a ring R_{Enc} of encodings, with a system of sets $\mathcal{S} = \{S_i^{(\alpha)} \subseteq R_{\text{Enc}} : \alpha \in R_{\text{Plain}}, 0 \leq i \leq \kappa\}$.



κ-Graded encoding scheme – key exchange

A ring R_{Plain} of plaintext and a ring R_{Enc} of encodings,
with a system of sets $\mathcal{S} = \{S_i^{(\alpha)} \subseteq R_{\text{Enc}} : \alpha \in R_{\text{Plain}}, 0 \leq i \leq \kappa\}$.



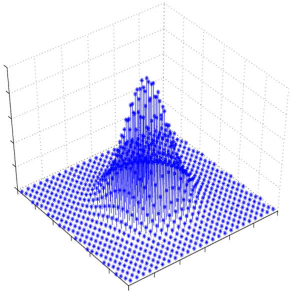
Notations

- Polynomial Ring: $R = \mathbb{Z}[x]/\langle x^n + 1\rangle$ for n power of 2,
- Let $q > 2$. We let $R_q = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$,
  - Arithmetic in R_q costs $\widetilde{O}(n \log q)$.

Discrete Gaussian on lattices

For a n-dimensional lattice Λ, a non-singular matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$:

$$\forall x \in \Lambda : \quad D_{\Lambda,\mathbf{S},c}[x] \sim \exp\left(-\pi\|\mathbf{S}^{-1}(x-c)\|^2\right).$$



- small size (depending on $\mathbf{S}$),
- sum is still Gaussian.

GGH

- **Instance generation** $\mathsf{InstGen}(1^\lambda, 1^\kappa)$:
  - Sample $g \hookleftarrow D_{R,\sigma}$ and $z \hookleftarrow U(R_q)$.
  - Sample a level-1 encoding of 1: $y = [a \cdot z^{-1}]_q$ with $a \hookleftarrow D_{1+\langle g \rangle, \sigma'}$.
  - For $i \leq \kappa$, sample m_r level-i encodings of 0: $(x_j^{(i)})_{j \leq m_r}$.
  - Return public parameters par $= (n, q, y, \{x_j^{(i)}\}_{j \leq m_r, i \leq \kappa})$.

- **Level-k encoding** $\mathsf{enc}_k(e)$: Given $e \in R/\langle g \rangle$:
  - Encode e at level k: Compute $u' = [e \cdot y^k]_q$ $(= [c'/z^k]_q)$.
  - Re-randomize: Sample $\rho_j \hookleftarrow D_{\mathbb{Z},\sigma^*}$ for $j \leq m_r$ and return

$$u = \left[u' + \sum_{j=1}^{m_r} \rho_j {x_j}^{(k)}\right]_q = \left[\left(c' + \sum_j \rho_j b_j^{(k)}\right)/z^k\right]_q$$

GGH

- **Instance generation** $\mathsf{InstGen}(1^\lambda, 1^\kappa)$:
  - Sample $g \hookleftarrow D_{R,\sigma}$ and $z \hookleftarrow U(R_q)$.
  - Sample a level-1 encoding of 1: $y = [a \cdot z^{-1}]_q$ with $a \hookleftarrow D_{1+\langle g \rangle,\sigma'}$.
  - For $i \leq \kappa$, sample m_r level-i encodings of 0: $(x_j^{(i)})_{j \leq m_r}$.
  - Return public parameters par $= (n, q, y, \{x_j^{(i)}\}_{j \leq m_r, i \leq \kappa})$.

- **Level-1 encoding** $\mathsf{enc}_1(e)$: Given $e \in R/\langle g \rangle$:
  - Encode e at level 1: Compute $u' = [e \cdot y]_q$ $(= [c'/z]_q)$.
  - Re-randomize: Sample $\rho_j \hookleftarrow D_{\mathbb{Z},\sigma^*}$ for $j \leq m_r$ and return

$$u = \left[u' + \sum_{j=1}^{m_r} \rho_j x_j\right]_q = \left[\left(c' + \sum_j \rho_j b_j\right)/z\right]_q$$

Encoding of zero

GGH

- **Instance generation** $\mathsf{InstGen}(1^\lambda, 1^\kappa)$:
  - Sample $g \hookleftarrow D_{R,\sigma}$ and $z \hookleftarrow U(R_q)$.
  - Sample a level-1 encoding of 1: $y = [a \cdot z^{-1}]_q$ with $a \hookleftarrow D_{1+\langle g \rangle, \sigma'}$.
  - For $i \leq \kappa$, sample m_r level-i encodings of 0: $(x_j^{(i)})_{j \leq m_r}$.
  - Return public parameters $\mathrm{par} = (n, q, y, \{x_j^{(i)}\}_{j \leq m_r, i \leq \kappa})$.

- **Level-1 encoding** $\mathsf{enc}_1(e)$: Given $e \in R/\langle g \rangle$:
  - Encode e at level 1: Compute $u' = [e \cdot y]_q$ $(= [c'/z]_q)$.
  - Re-randomize: Sample $\rho_j \hookleftarrow D_{\mathbb{Z},\sigma^*}$ for $j \leq m_r$ and return

$$u = \left[u' + \sum_{j=1}^{m_r} \rho_j x_j\right]_q = \left[\left(c' + \sum_j \rho_j b_j\right)/z\right]_q$$

Encoding of zero

Discrete Gaussian over $\mathbb{Z}$

Security

Which security?

- **Graded Decisional Diffie-Hellman** – GDDH:
  Given $\kappa + 1$ level-1 encoding u_i of plaintexts e_i, distinguish between a level-κ encoding of the product of the e_i and a level-κ encoding of a random element.

To ensure security $\Rightarrow$ need randomization of the encodings

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

With which parameters?

This work: understand the security of the re-randomization and propose efficient GGH variant achieving this security.

GGHLite: Our contribution

We improve encoding re-randomization in GGH

- m_r level-1 encodings of 0: $\{x_j\}_{j \leq m_r}$,
- To randomize $u' = [e \cdot y]_q$, output $u = [u' + \sum_j \rho_j b_j / z]_q$,
- Randomizers ρ_j's are sampled from $D_{\mathbb{Z},\sigma^*} \Rightarrow \sum_j \rho_j b_j$ Gaussian.

GGHLite: Our contribution

We improve encoding re-randomization in GGH

- m_r level-1 encodings of 0: $\{x_j\}_{j \leq m_r}$,
- To randomize $u' = [e \cdot y]_q$, output $u = [u' + \sum_j \rho_j b_j / z]_q$,
- Randomizers ρ_j's are sampled from $D_{\mathbb{Z},\sigma^*} \Rightarrow \sum_j \rho_j b_j$ Gaussian.

By:

- Formalizing the security goal,
    - Introduction of a **canonical version** of the problem.

GGHLite: Our contribution

We improve encoding re-randomization in GGH

- m_r level-1 encodings of 0: $\{x_j\}_{j \leq m_r}$,
- To randomize $u' = [e \cdot y]_q$, output $u = [u' + \sum_j \rho_j b_j / z]_q$,
- Randomizers ρ_j's are sampled from $D_{\mathbb{Z},\sigma^*} \Rightarrow \sum_j \rho_j b_j$ Gaussian.

By:

- Formalizing the security goal,
  - Introduction of a **canonical version** of the problem.
- Decreasing the size of σ^*,
  - Reduction from the canonical version to GCDH using **Rényi divergence** instead of the statistical distance.

GGHLite: Our contribution

We improve encoding re-randomization in GGH

- m_r level-1 encodings of 0: $\{x_j\}_{j \leq m_r}$,
- To randomize $u' = [e \cdot y]_q$, output $u = [u' + \sum_j \rho_j b_j / z]_q$,
- Randomizers ρ_j's are sampled from $D_{\mathbb{Z},\sigma^*} \Rightarrow \sum_j \rho_j b_j$ Gaussian.

By:

- Formalizing the security goal,
  - Introduction of a **canonical version** of the problem.
- Decreasing the size of σ^*,
  - Reduction from the canonical version to GCDH using **Rényi divergence** instead of the statistical distance.
- Decreasing the number m_r of re-randomizers.
  - New **Leftover Hash Lemma**.

GGHLite: Formalizing Re-randomization Security

- **Informal requirement:** Prevent statistical correlation between re-randomized encoding and encoded element.
- **Formal requirement:** Breaking GCDH problem is as hard as breaking canonical GCDH problem.

B has columns the b_j's

We define:

GCDH:	canonical **GCDH**:
Given $u_i = [(c'_i + \sum_{j=1}^{m_r} \rho_{j,i} \cdot b_j) z^{-1}]_q = [\ c_i\ z^{-1}]_q = \mathsf{Enc}_1(e_i)$	Given $u_i = [\ c_i\ z^{-1}]_q$ with $c_i \hookleftarrow D_{\langle g \rangle + e_i, \sigma^* B^T, 0}$

$\Rightarrow$ compute level κ encoding of the product $c_1 \cdots c_{\kappa+1}$.

$c_i \approx D_{\langle g \rangle + e_i, \sigma^* B^T, c'_i}$ small centre c'_i.	$c_i \approx D_{\langle g \rangle + e_i, \sigma^* B^T, 0}$ zero centre.

GGHLite Re-randomization Security: First Ingredient

Distribution of c_i (with $u_i = [c_i/z]_q$):

GCDH	**can-GCDH**
$D_1 \approx D_{\langle g\rangle + e_i, \sigma^* B^T, c'_i}$	$D_2 \approx D_{\langle g\rangle + e_i, \sigma^* B^T, 0}$
small centre c'_i.	zero centre.

GGH security reduction based on statistical distance (SD):

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

Adversary $\mathcal{A}_{\mathbf{GCDH}}$ with success $\varepsilon \Rightarrow \mathcal{A}_{can\mathbf{GCDH}}$ with success ε'

$$\varepsilon' \geq \varepsilon - \Delta(D_1, D_2),$$

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

GGHLite Re-randomization Security: First Ingredient

Distribution of c_i (with $u_i = [c_i/z]_q$):

GCDH	**can-GCDH**
$D_1 \approx D_{\langle g \rangle + e_i, \sigma^* B^T, c'_i}$	$D_2 \approx D_{\langle g \rangle + e_i, \sigma^* B^T, 0}$
small centre c'_i.	zero centre.

GGHLite security reduction based on Rényi divergence (RD):

$$R(D_1 \| D_2) \stackrel{\text{def}}{=} \sum_x \frac{D_1^2(x)}{D_2(x)},$$

Adversary $\mathcal{A}_{\mathbf{GCDH}}$ with success $\varepsilon \Rightarrow \mathcal{A}_{can\mathbf{GCDH}}$ with success ε'

$$\varepsilon' \geq \frac{\varepsilon^2}{R(D_1 \| D_2)},$$

- Useful even if $\varepsilon < R(D_1, D_2)^{-1}$ – use $R(D_1 \| D_2) \leq \text{poly}(\lambda)$.
- For $R(D_1 \| D_2) \leq \text{poly}(\lambda)$, can use $\frac{\sigma^*}{\|c'_i\|} = O(\frac{1}{\log \lambda})$.

GGHLite: Second Main Ingredient

New Leftover Hash Lemma

In GGH construction:

- Needs $m_r = \Omega(n \log n)$ encodings of 0,
- Uses rational integer Gaussian randomizers ($\rho_j \in \mathbb{Z}$),
- Uses a discrete Gaussian Leftover Hash Lemma (LHL) to show $\sum_{j \leq m_r} \rho_j b_j$ distribution is close to a discrete Gaussian on $\langle g \rangle$.

GGHLite second ingredient: $m_r = 2$ encodings of 0 suffice

- Uses Gaussian randomizers over full ring ($\rho_j \in R$),
- New algebraic variant of discrete Gaussian LHL over R: $\sum_{j \leq m_r} \rho_j b_j$ distribution is close to a discrete Gaussian on $\langle g \rangle$.

GGHLite: Asymptotic Parameters

For κ level:

Parameter	GGHLite	GGH
m_r	2	$\Omega(n \log n)$
σ^*	$\widetilde{O}(n^{5.5}\sqrt{\kappa})$	$\widetilde{O}(2^{\lambda}\lambda n^{4.5}\kappa)$
q	$\widetilde{O}((n^{10.5}\sqrt{\kappa})^{8\kappa})$	$\widetilde{O}((2^{\lambda}\lambda^{1.5}n^{8.5}\kappa)^{8\kappa})$
n	$O(\kappa\lambda\log\lambda)$	$O(\kappa\lambda^2)$
$\lvert\mathsf{enc}\rvert$	$O(\kappa^2\lambda\log^2\lambda)$	$O(\kappa^2\lambda^3)$
$\lvert\mathrm{par}\rvert$	$O(\kappa^3\lambda\log^2\lambda)$	$O(\kappa^4\lambda^5\log\lambda)$

Adapting Applications of GGH to GGHLite

Question: How to adapt GGH applications to rely on GCDH rather than GDDH?

Answer: Replace $K = v$ in original protocol by

$$K = H(v)$$

in modified protocol, where $H(\cdot)$ is a cryptographic hash function.

If $H(\cdot)$ is modelled as a Random Oracle Model, then security of modified protocol relies on GCDH – our GGHLite analysis applies.

Conclusion

GGHLite: a more efficient variant of
GGH graded encoding scheme.

- Reduction from can-**GCDH** to **GCDH** using Rényi divergence,
- New algebraic variant of discrete Gaussian LHL over R.

Open Problems:

- Can our Rényi divergence analysis be applied to the Decision Graded Diffie Hellman problem?
- Understand the complexity of canonical GCDH problem – provable relation to standard lattice problems?
- Understand relation between GGH/GGHLite and more recent Jigsaw puzzle variants (obfuscation).
- Concrete computational / space efficiency of GGHLite based on best known attacks?