Multilinear Maps over the Integers

Jean-Sébastien Coron, Tancrède Lepoint and Mehdi Tibouchi

University of Luxembourg, CryptoExperts, NTT Secure Platform Laboratories

June 5th, 2015

Motivation

$\hat{t} : G_1 \times G_2 \to G_T$

- Bilinear maps (from pairing in hard-DL groups) are extremely useful in cryptography
  - 3-partite Diffie-Hellman key exchange
  - *-BE (IBE, HIBE, ABE, etc.)
  - NIZK proofs, Traitor Tracing, broadcast encryption, etc.

Motivation

$\hat{t} : G_1 \times G_2 \rightarrow G_T$

- Bilinear maps (from pairing in hard-DL groups) are extremely useful in cryptography
  - 3-partite Diffie-Hellman key exchange
  - *-BE (IBE, HIBE, ABE, etc.)
  - NIZK proofs, Traitor Tracing, broadcast encryption, etc.
- What could we do with multilinear maps?
  - 2003 Boneh and Silverberg: N-multipartite Diffie-Hellman and very efficient broadcast encryption
  - Certainly a lot…
  - … but pessimistic about existence of such maps in the realm of algebraic geometry!

[GGH13]: First Multilinear Maps Candidate

- Garg, Gentry and Halevi breakthrough in 2012
  - First plausible candidate of Multilinear Maps
  - Not exactly generalization of bilinear maps
  - But introduction of noisy encodings and Graded Encoded Systems
  - Based on ideal lattices & ideas similar to NTRU
  - Published at Eurocrypt 2013 [GGH13]
- New construction similarly flavored as FHE
- Useful for applications: e.g. description of a N-multipartite Diffie Hellman key exchange protocol

[GGH13]: First Multilinear Maps Candidate

- Garg, Gentry and Halevi breakthrough in 2012
  - First plausible candidate of Multilinear Maps
  - Not exactly generalization of bilinear maps
  - But introduction of noisy encodings and Graded Encoded Systems
  - Based on ideal lattices & ideas similar to NTRU
  - Published at Eurocrypt 2013 [GGH13]
- New construction similarly flavored as FHE
- Useful for applications: e.g. description of a N-multipartite Diffie Hellman key exchange protocol
- Broken by Hu and Jia on March 2015 (ePrint)

Following [GGH13] Multilinear Map Breakthrough

- Witness Encryption (STOC 2013)
- Full Domain Hash and Identity-Based Aggregate Signatures (CRYPTO 2013)
- Programmable hash functions (CRYPTO 2013)
- ABE for circuits (CRYPTO 2013)
- Obfuscation (CRYPTO 2013 + FOCS 2013 + Eprint)

Following [GGH13] Multilinear Map Breakthrough

- Witness Encryption (STOC 2013)
- Full Domain Hash and Identity-Based Aggregate Signatures (CRYPTO 2013)
- Programmable hash functions (CRYPTO 2013)
- ABE for circuits (CRYPTO 2013)
- Obfuscation (CRYPTO 2013 + FOCS 2013 + Eprint)

- GGHLite: more efficient multilinear maps from ideal lattices (Eurocrypt 2014)
  - Variant of GGH with much small public parameters.
  - Still broken by Hu and Jia's attack.

Multilinear Maps over the Integers

- Published at Crypto 2013 by Coron, Lepoint, and Tibouchi (CLT).
- Similar approach as [GGH13] but over the integers instead of ideal lattices.
- as in the DGHV fully homomorphic encryption scheme

Multilinear Maps over the Integers

- Published at Crypto 2013 by Coron, Lepoint, and Tibouchi (CLT).
- Similar approach as [GGH13] but over the integers instead of ideal lattices.
- as in the DGHV fully homomorphic encryption scheme

- First implementation of a 7-multipartite Diffie-Hellman (80-bit sec.)
  - Public parameters (shared): 2.5GB
  - Arguably reasonable timings for key agreement: $\leqslant$ 40s/user

Multilinear Maps over the Integers

- Published at Crypto 2013 by Coron, Lepoint, and Tibouchi (CLT).
- Similar approach as [GGH13] but over the integers instead of ideal lattices.
- as in the DGHV fully homomorphic encryption scheme

- First implementation of a 7-multipartite Diffie-Hellman (80-bit sec.)
  - Public parameters (shared): 2.5GB
  - Arguably reasonable timings for key agreement: $\leqslant$ 40s/user

- Broken by Cheon *et al.* at Eurocrypt 2015.

- But: can still be used for obfuscation.

Recall: Bilinear Maps

- Two groups and a mapping $e \colon G_1 \times G_1 \to G_2$
  - Groups written multiplicatively
  - Bilinear: $e(x^a, y^b) = e(x, y)^{ab}$

Recall: Bilinear Maps

- Two groups and a mapping $e\colon G_1 \times G_1 \to G_2$
  - Groups written multiplicatively
  - Bilinear: $e(x^a, y^b) = e(x, y)^{ab}$
- Hard problems

  - DL: Given (g, g^a), find a
  - DH: Given (g, g^a, g^b), find g^{ab}
  - BDH: Given (g, g^a, g^b, g^c), compute $e(g, g)^{abc}$

Recall: Bilinear Maps

- Two groups and a mapping $e\colon G_1 \times G_1 \to G_2$
  - Groups written multiplicatively
  - Bilinear: $e(x^a, y^b) = e(x, y)^{ab}$
- Hard problems

  - DL: Given (g, g^a), find a
  - DH: Given (g, g^a, g^b), find g^{ab}
  - BDH: Given (g, g^a, g^b, g^c), compute $e(g, g)^{abc}$
- Application: non-interactive 3-party Diffie-Hellman (Joux, 2000)

$$sk = e(g^a, g^b)^c = e(g^a, g^c)^b = e(g^b, g^c)^a$$

Extension to multilinear map

- Bilinear pairings: $a \in \mathbb{Z}_p \mapsto g^a$ is an “encoding” of the scalar a
  - easy to encode, hard to decode (DL)
  - additively and multiplicatively homomorphic
    - from g^a, g^b, compute g^{a+b} and $e(g,g)^{ab}$

Extension to multilinear map

- Bilinear pairings: $a \in \mathbb{Z}_p \mapsto g^a$ is an "encoding" of the scalar a
  - easy to encode, hard to decode (DL)
  - additively and multiplicatively homomorphic
    - from g^a, g^b, compute g^{a+b} and $e(g,g)^{ab}$
- It would be interesting to have a κ-linear map

$$e \colon G_1 \times G_2 \times \cdots \times G_\kappa \to G_{\kappa+1}$$

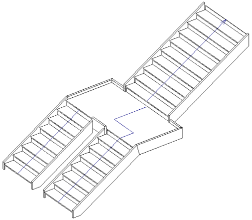
- Application: non-interactive Diffie-Hellman key exchange with $\kappa + 1$ users.

$$sk = e(g^{a_1}, \ldots, g^{a_\kappa})^{a_{\kappa+1}}$$

Perspective of [GGH13]

- Perfect multilinear map $e\colon G_1 \times G_2 \times \cdots \times G_k \to G_{k+1}$
- Cannot really do that... but slightly analogous:
  - The one-way $g \mapsto g^a$ is replaced by *randomized* encodings ($a \in R$ has many encodings)
  - final multilinear map $e(g^{a_1}, \ldots, g^{a_\kappa})$ has a deterministic part depending on the a_i's only
  - The multilinear map is essentially a homomorphic multiplication of these encodings, followed by an operation that deterministically extracts some bits from the product

Perspective of [GGH13]: Graded Encoding



- Each encoding is associated to a level
  - level-0: "plaintext" scalars $a \in R$
  - level-1: encoding g^a
  - level-κ: by combining κ level-1 encoding
  - we can multiply any bounded subset of encodings until level κ
  - at level κ, special "zero-testing" element which can extract a deterministic function of ring elements
- Public parameters hide secret information

The CLT2013 encoding scheme

- Parameters: sec. level λ, multilinearity level κ
- Public modulus: $x_0 = p_1 \times \cdots \times p_n$ where p_i primes
- Random secret mask: $z \in (\mathbb{Z}/x_0\mathbb{Z})^\times$

- Level-k encoding of $\mathbf{m} = (m_i) \in R := (\mathbb{Z}/g_1\mathbb{Z}) \times \cdots \times (\mathbb{Z}/g_n\mathbb{Z})$:

$$c \equiv \frac{r_i \cdot g_i + m_i}{z^k} \bmod p_i$$

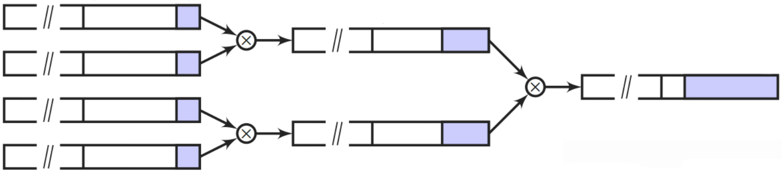
- Multiplicative mask z (used for extraction at level κ)

CLT2013: homomorphic properties

- Level-k encoding of $\mathbf{m} = (m_i) \in R := (\mathbb{Z}/g_1\mathbb{Z}) \times \cdots \times (\mathbb{Z}/g_n\mathbb{Z})$:

$$c \equiv \frac{r_i \cdot g_i + m_i}{z^k} \bmod p_i$$

- Additively and Multiplicatively Homomorphic
  - After addition: noise $r' \approx 2 \max r_i$
  - After multiplication: noise $r' \approx \max g_i \cdot r_i^2$



The CLT2013 encoding scheme

- But how can we generate a level-k encoding?

$$c \equiv \frac{r_i \cdot g_i + m_i}{z^k} \bmod p_i$$

The CLT2013 encoding scheme

- But how can we generate a level-k encoding? ... we should compute a CRT:

$$c = \mathrm{CRT}_{p_1,\ldots,p_n}\left(\frac{r_1 \cdot g_1 + m_1}{z^k}, \ldots, \frac{r_n \cdot g_n + m_n}{z^k}\right)$$

The CLT2013 encoding scheme

- But how can we generate a level-k encoding? ... we should compute a CRT:

$$c = \mathrm{CRT}_{p_1,\ldots,p_n}\left(\frac{r_1 \cdot g_1 + m_1}{z^k}, \ldots, \frac{r_n \cdot g_n + m_n}{z^k}\right)$$

- But we do not know the p_i's and z...

The CLT2013 encoding scheme

- But how can we generate a level-k encoding? ... we should compute a CRT:

$$c = \mathrm{CRT}_{p_1,\ldots,p_n}\left(\frac{r_1 \cdot g_1 + m_1}{z^k}, \ldots, \frac{r_n \cdot g_n + m_n}{z^k}\right)$$

- But we do not know the p_i's and z...

- In the protocols we need to generate random encodings
- $\Rightarrow$ we generate using random subset sum of public encodings and apply the left-over hash lemma

Public generation of Level-0 Encoding

- Publish ξ_i: level-0 encodings of random elements in R.
- **Ring Sampler:** random subset sum of $\xi_i \Rightarrow$ level-0 encoding of random element in R (uses LHL)

$$c_0 = \sum_{i \in S} \xi_i \bmod x_0$$

Public generation of Level-0 Encoding

- Publish ξ_i: level-0 encodings of random elements in R.
- **Ring Sampler:** random subset sum of $\xi_i \Rightarrow$ level-0 encoding of random element in R (uses LHL)

$$c_0 = \sum_{i \in S} \xi_i \bmod x_0$$

- Publish y: level-1 encoding of $\mathbf{1} = (1, \ldots, 1) \in R$
- **Encoding:** Transform it into a level-1 encoding by multiplication by y

$$c_1 = (c_0 \cdot y) \bmod x_0$$

Public generation of Level-0 Encoding

- Publish ξ_i: level-0 encodings of random elements in R.
- **Ring Sampler:** random subset sum of $\xi_i \Rightarrow$ level-0 encoding of random element in R (uses LHL)

$$c_0 = \sum_{i \in S} \xi_i \bmod x_0$$

- Publish y: level-1 encoding of $\mathbf{1} = (1, \ldots, 1) \in R$
- **Encoding:** Transform it into a level-1 encoding by multiplication by y

$$c_1 = (c_0 \cdot y) \bmod x_0$$

- Publish x_i: level-1 encodings of 0.
- **Re-Randomization:** Re-randomize your element c_1 with a subset sum of the x_i's

$$c_1' = c_1 + \sum_{i \in S'} x_i$$

Diffie-Hellman key exchange

- Every user generates a pair (a_i, u_i)
  - a_i is a level-0 encoding of a random (unknown) $\alpha_i \in R$.
  - u_i is a level-1 encoding of the same $\alpha_i \in R$
  - Public: u_i. Private: a_i.

Diffie-Hellman key exchange

- Every user generates a pair (a_i, u_i)
  - a_i is a level-0 encoding of a random (unknown) $\alpha_i \in R$.
  - u_i is a level-1 encoding of the same $\alpha_i \in R$
  - Public: u_i. Private: a_i.
- Why re-randomization ?
  - Without re-rerandomization, one could compute $a_i = u_i / y \bmod x_0$.

Diffie-Hellman key exchange

- Every user generates a pair (a_i, u_i)
  - a_i is a level-0 encoding of a random (unknown) $\alpha_i \in R$.
  - u_i is a level-1 encoding of the same $\alpha_i \in R$
  - Public: u_i. Private: a_i.
- Why re-randomization ?
  - Without re-rerandomization, one could compute $a_i = u_i / y \bmod x_0$.
- Diffie-Hellman key exchange with $\kappa + 1$ users. User i can compute:

$$c_i = a_i \cdot \prod_{j \neq i} u_j \bmod x_0$$

  - c_i is a level-κ (randomized) encoding of $\alpha = \prod_i \alpha_i$
  - An attacker cannot compute such level-κ encoding
  - Each user should be able to extract from c_i the same secret-key that depends only on α.

Zero-Tester: How to Extract Deterministically?

- Publish a zero-testing element to check that level-κ encodings are encodings of $\mathbf{0}$:

$$p_{zt} = \sum_{i=1}^{n} h_i \cdot (z^\kappa \cdot g_i^{-1} \bmod p_i) \cdot \prod_{i' \neq i} p_{i'} \bmod x_0$$

- Compute $\omega = p_{zt} \cdot c \bmod x_0$
- Extract the MSB of ω: if $\mathbf{m} = 0$ then the MSB are 0.
  - $\Rightarrow$ MSB of two encodings of the same $\mathbf{m}$ are the same (e.g. derive common session key)

Zero-Tester: How to Extract Deterministically?

- Publish a zero-testing element to check that level-κ encodings are encodings of $\mathbf{0}$:

$$p_{zt} = \sum_{i=1}^{n} h_i \cdot (z^{\kappa} \cdot g_i^{-1} \bmod p_i) \cdot \prod_{i' \neq i} p_{i'} \bmod x_0$$

- Compute $\omega = p_{zt} \cdot c \bmod x_0$
- Extract the MSB of ω: if $\mathbf{m} = 0$ then the MSB are 0.
  - $\Rightarrow$ MSB of two encodings of the same $\mathbf{m}$ are the same (e.g. derive common session key)

Main Idea: what happens if we work with only one p_1?

- $c \equiv \frac{r_1 \cdot g_1 + m_1}{z^{\kappa}} \pmod{p_1}$ and $p_{zt} \equiv \frac{h_1 \cdot z^{\kappa}}{g_1} \pmod{p_1}$

Zero-Tester: How to Extract Deterministically?

- Publish a zero-testing element to check that level-κ encodings are encodings of $\mathbf{0}$:

$$p_{zt} = \sum_{i=1}^{n} h_i \cdot (z^\kappa \cdot g_i^{-1} \bmod p_i) \cdot \prod_{i' \neq i} p_{i'} \bmod x_0$$

- Compute $\omega = p_{zt} \cdot c \bmod x_0$
- Extract the MSB of ω: if $\mathbf{m} = 0$ then the MSB are 0.

  $\Rightarrow$ MSB of two encodings of the same $\mathbf{m}$ are the same (e.g. derive common session key)

Main Idea: what happens if we work with only one p_1?

- $c \equiv \frac{r_1 \cdot g_1 + m_1}{z^\kappa} \pmod{p_1}$ and $p_{zt} \equiv \frac{h_1 \cdot z^\kappa}{g_1} \pmod{p_1}$

$\Rightarrow$ $$\omega \equiv p_{zt} \cdot c \equiv h_1 r_1 + h_1 \frac{m_1}{g_1} \pmod{p_1}$$

Zero-Tester: How to Extract Deterministically?

- Publish a zero-testing element to check that level-κ encodings are encodings of $\mathbf{0}$:

$$p_{zt} = \sum_{i=1}^{n} h_i \cdot (z^\kappa \cdot g_i^{-1} \bmod p_i) \cdot \prod_{i' \neq i} p_{i'} \bmod x_0$$

- Compute $\omega = p_{zt} \cdot c \bmod x_0$
- Extract the MSB of ω: if $\mathbf{m} = 0$ then the MSB are 0.

  $\Rightarrow$ MSB of two encodings of the same $\mathbf{m}$ are the same (e.g. derive common session key)

Main Idea: what happens if we work with only one p_1?

- $c \equiv \frac{r_1 \cdot g_1 + m_1}{z^\kappa} \pmod{p_1}$ and $p_{zt} \equiv \frac{h_1 \cdot z^\kappa}{g_1} \pmod{p_1}$

$\Rightarrow$
$$\omega \equiv p_{zt} \cdot c \equiv h_1 r_1 + h_1 \frac{m_1}{g_1} \pmod{p_1}$$

- When $m_1 = 0$, $\omega = h_1 \cdot r_1 \ll p_1 \Rightarrow$ all its MSB are equal to 0

Zero-testing parameter: deterministic extraction

- Level-κ encoding:

$$c \equiv \frac{r_i \cdot g_i + m_i}{z^k} \pmod{p_i}$$

- Zero-testing parameter:

$$p_{zt} = \sum_{i=1}^{n} h_i \cdot [g_i^{-1} \cdot z^{\kappa} \bmod p_i] \cdot (x_0/p_i) \bmod x_0$$

- Extraction from $\omega = p_{zt} \cdot c \bmod x_0$:

$$\omega = \sum_{i=1}^{n} h_i \cdot \left(r_i + m_i \cdot (g_i^{-1} \bmod p_i)\right) \cdot (x_0/p_i) \bmod x_0$$

- The MSB of ω only depends on the m_i's (for small r_i's and h_i's).

The Cheon *et al.* Attack (Eurocrypt 2015)

- Total break of the CLT2013 scheme: recover in polynomial time all secret parameters.
- Uses a large number of low-level encodings of 0
  - Let x'_j be level-1 encodings of $0 \in R$ with $x'_j = r'_{ij} \cdot g_i / z \bmod p_i$
  - Let x_j be level-1 encodings where $x_j = x_{ij}/z \bmod p_i$.
- For $1 \leqslant j, k \leqslant n$, compute:

$$\omega_{jk} = (c \cdot x_j \cdot x'_k \cdot y^{\kappa-2}) \cdot p_{zt} \bmod x_0$$

where c is a level-0 encoding with $c = c_i \bmod p_i$.

$$\omega_{jk} = \sum_{i=1}^{n} h_i \cdot [(c \cdot x_j \cdot x'_k \cdot y^{\kappa-2}) \cdot z^{\kappa} \cdot g_i^{-1} \bmod p_i] \cdot (x_0/p_i)$$

$$= \sum_{i=1}^{n} x_{ij} h'_i c_i r'_{ik} \bmod x_0 \qquad (1)$$

- Equation (1) actually holds over the integers.

The Cheon *et al.* Attack

- For all $1 \leqslant j, k \leqslant n$, ω_{jk} is a quadratic form in x_{ij} and r'_{ik}:

$$\omega_{jk} = \sum_{i=1}^{n} x_{ij} h'_i c_i r'_{ik}$$

- In matrix form with $\mathbf{W}_c = (\omega_{jk})_{1 \leqslant j,k \leqslant n}$:

$$\mathbf{W}_c = \mathbf{X} \times \mathbf{C} \times \mathbf{R}\,,$$

  where $\mathbf{X} = (x_{ij} \cdot h'_i)_{1 \leqslant j,i \leqslant n}$ and $\mathbf{R} = (r'_{ik})_{1 \leqslant i,k \leqslant n}$ and $\mathbf{C} = \mathrm{Diag}(c_1, \ldots, c_n)$.

- Compute with $c = 1$ the matrix $\mathbf{W_1} = \mathbf{X} \times \mathbf{I} \times \mathbf{R}$, which gives:

$$\mathbf{W} = \mathbf{W}_c \cdot {\mathbf{W}_1}^{-1} = \mathbf{X} \times \mathbf{C} \times \mathbf{X}^{-1}$$

- Compute the eigenvalues of $\mathbf{W}$ and recover the c_i's, and eventually the p_i's.

Extension of Cheon *et al.* Attack

- [GGHZ14] countermeasure
  - Embed CLT encodings into a matrix of encodings
  - Eliminate native encodings of 0 to prevent the Cheon *et al.* attack.
- [BWZ14] countermeasure
  - Uses pairs of encodings
  - Also eliminate native encodings of 0 to prevent the Cheon *et al.* attack.
- Both countermeasures can be broken by a simple extension of the Cheon *et al.* attack
  - Namely in both case ω is still a quadratic form.
  - Therefore Cheon *et al.* attack applies with higher-dimensional matrices.
- "Zeroizing Without Low-level Zeroes: New Attacks on Multilinear Maps and Their Limitations", to appear at Crypto 2015