UNIVERSITÉ DE RENNES 1

SEOUL NATIONAL UNIVERSITY

Cryptanalysis of the CLT15 Multilinear Map over the Integers

Jung Hee Cheon, Pierre-Alain Fouque, Changmin Lee, *Brice Minaud*, Hansol Ryu

ENS Lyon, January 19th 2017

Plan

1. Multilinear Maps.
2. The CLT15 Multilinear Map.
3. Cryptanalysis.

Conclusion.

Multilinear Maps

Multilinear Maps

- Powerful cryptographic primitive.

  Generalization of pairings.

  Introduced in 2002 [BS02].

  First construction(s) in 2013 [GGH13, CLT13].

- Numerous applications, "crypto-complete".

  Non-interactive multipartite key exchange (direct application), witness encryption...

  Indistinguishability obfuscation [GGHRSW13].

Candidate Schemes

Few actual schemes.

- GGH13 (on lattices)
  Garg, Gentry, Halevi, Eurocrypt'13 ✘ [HJ15] (key exch.)
- CLT13 (on integers)
  Coron, Lepoint, Tibouchi, Crypto'13 ✘ [CHLRS15] (idem)
- GGH15 (graph-based)
  Gentry, Gorbunov, Halevi, TCC'15 ✘ [CLLT16] (idem)
- CLT15 (modified CLT13)
  Coron, Lepoint, Tibouchi, Crypto'15 ✘ [CFLMR16] (idem)

Definition

Multilinear Maps

Message: $c \in \mathbb{Z}/n\mathbb{Z}$

Encoding: $g^c \in \mathbb{G}$ $\quad$ $\mathbb{G}$ group of order n generated by g.

Additive homomorphism :

Addition of messages = multiplication of encodings. ✔

$$g^a g^b = g^{a+b}$$

Multiplication of messages = Diffie-Hellman. ✘

$$(g^a, g^b) \mapsto g^{ab} \ ?$$

Multilinear Maps

κ-**Multilinear Map** (symmetric case):

$$e : \mathbb{G}^{\kappa} \to \mathbb{H}$$
$$(g^{x_1}, g^{x_2}, \ldots, g^{x_\kappa}) \mapsto h^{x_1 \cdots x_\kappa}$$

where g, h are generators of $\mathbb{G}$, $\mathbb{H}$.

e.g. a 2-multilinear map is a pairing.

Non-Interactive Key Exchange

Assume we have a 3-multilinear map.

Then we can do 4-party non-interactive key exchange.

User	Draws	Publishes	Computes
A	a	g^a	$h^{abcd} = e(g^b, g^c, g^d)^a$
B	b	g^b	$h^{abcd} = e(g^a, g^c, g^d)^b$
C	c	g^c	$h^{abcd} = e(g^a, g^b, g^d)^c$
D	d	g^d	$h^{abcd} = e(g^a, g^b, g^c)^d$

Security: cannot compute h^{abcd} from g^a, g^b, g^c, g^d.

Leveled Multilinear Maps

Multilinear Map :

$$e : \mathbb{G}^{\kappa} \to \mathbb{H}$$
$$(g^{x_1}, g^{x_2}, \ldots, g^{x_\kappa}) \mapsto h^{x_1 \cdots x_\kappa}$$

where g, h are generators of $\mathbb{G}$, $\mathbb{H}$.

Leveled **Multilinear Map** :

$$e_{i,j} : \mathbb{G}_i \times \mathbb{G}_j \to \mathbb{G}_{i+j} \qquad \text{for } i + j \leq \kappa.$$
$$(g_i^{x}, g_j^{y}) \mapsto g_{i+j}^{xy}$$

where g_i is a generator of $\mathbb{G}_i$, $i \leq \kappa$.

Graded Encoding Schemes

Graded Encoding Scheme :

Message : $c \in \mathcal{P}$

Encoding : $\mathrm{enc}_i(c) \in \mathcal{C}_i$ et level *i*. Non-deterministic.

Encodings satisfy :

- Addition : $\mathrm{enc}_i(x) + \mathrm{enc}_i(y) = \mathrm{enc}_i(x+y)$
- Multiplication : $\mathrm{enc}_i(x) \cdot \mathrm{enc}_j(y) = \mathrm{enc}_{i+j}(xy)$

Encodings are noisy (*à la* FHE).

Zero-testing : public mapping $\mathrm{z} : \mathcal{C}_\kappa \to \{0,1\}$

$$\mathrm{enc}_\kappa(x) \mapsto 1 \text{ ssi } x = 0$$

Graded Encoding Schemes

Graded Encoding Scheme :

Message : $c \in \mathcal{P}$

Encoding : $\mathrm{enc}_i(c) \in \mathcal{C}_i$ et level *i*. Non-deterministic.

Encodings satisfy :

- Addition : $\mathrm{enc}_i(x) + \mathrm{enc}_i(y) = \mathrm{enc}_i(x + y)$
- Multiplication : $\mathrm{enc}_i(x) \cdot \mathrm{enc}_j(y) = \mathrm{enc}_{i+j}(xy)$

Encodings are noisy (*à la* FHE).

Extraction : public mapping $\mathrm{ext} : \mathcal{C}_\kappa \to \{0, 1\}$

$$\mathrm{enc}_\kappa(x) \mapsto H(x)$$

Non-Interactive Key Exchange v2

Assume we have a **3**-graded encoding scheme.

User	Draws	Publishes	Computes
A	***a***	$enc_1(a)$	ext($\boldsymbol{a} \cdot enc_3(bcd)$)
B	***b***	$enc_1(b)$	ext($\boldsymbol{b} \cdot enc_3(acd)$)
C	***c***	$enc_1(c)$	ext($\boldsymbol{c} \cdot enc_3(abd)$)
D	***d***	$enc_1(d)$	ext($\boldsymbol{d} \cdot enc_3(abc)$)

Public key contains $enc_1(1)$, many instances $enc_0(\$)$, $enc_1(0)$.

Security: cannot compute ext($enc_4(abcd)$) from $\mathrm{enc}_1(a)$, $\mathrm{enc}_1(b)$, $\mathrm{enc}_1(c)$, $\mathrm{enc}_1(d)$.

CLT15 Multilinear Map

Encoding in CLT15

Let g_i and p_i denote n prime numbers with $g_i \ll p_i$.
Let $z < x_0 = \prod p_i$.

Message space : $\prod_{i \leq n} \mathbb{Z}/g_i\mathbb{Z}$

Encoding of $(m_1, \ldots, m_n) \in \prod_{i \leq n} \mathbb{Z}/g_i\mathbb{Z}$ at level k :

integer e such that $\forall i, e \bmod p_i = \dfrac{r_i g_i + m_i}{z^k} \bmod p_i$:

$$e = \mathrm{CRT}_{(p_i)}\left(\frac{r_i g_i + m_i}{z^k}\right) + a x_0$$

with r_i, a, small (secret) noise.

(biggest diff with CLT13)

Operations in CLT15

Encoding at level k : $e = \mathrm{CRT}_{(p_i)}\left(\frac{r_i g_i + m_i}{z^k}\right) + a x_0$

Addition and multiplication of encodings
= addition and multiplication over the integers !

e.g. $\mathrm{CRT}_{(p_i)}\left(\frac{r_i g_i + m_i}{z^k}\right) + \mathrm{CRT}_{(p_i)}\left(\frac{r'_i g_i + m'_i}{z^k}\right)$

$= \mathrm{CRT}_{(p_i)}\left(\frac{(r_i + r'_i) g_i + m_i + m'_i}{z^k}\right)$

...as long as reduction mod p_i does not interfere, i.e.:

$$|r_i g_i + m_i| \ll p_i$$

...other caveat: multiplication doubles the size of encodings.

Reduction Ladder

- Solution: public key contains a **ladder** of encodings of zero of increasing size. That is, encodings $\{X_i : i \leq m\}$ of zero with:

$$\text{size}(X_0) = \text{size}(x_0) + 2\rho \quad \textit{largest size allowed for encoding.}$$

$$\text{size}(X_1) = \text{size}(X_0) + 1$$

$$\text{size}(X_2) = \text{size}(X_0) + 2$$

$$\dots$$

$$\text{size}(X_m) = 2\text{size}(X_0) \quad \textit{largest size possible for product.}$$

- **Reduce** an encoding =
  - Substract largest possible ladder element.
  - Repeat until $< X_0$.

Zero-testing in CLT15

$$\text{Encoding at level } \kappa : e = \text{CRT}_{(p_i)}\left(\frac{r_i g_i + m_i}{z^\kappa}\right) + a x_0$$

$$\text{Development} : e = \sum (r_i + m_i {g_i}^{-1}) u_i + a x_0$$

Zero-testing : prime $N \gg x_0$, integer $p_{zt} < N$ such that :

$$|v_0| = |x_0 p_{zt} \bmod N| \ll N$$

$$|v_i| = |u_i p_{zt} \bmod N| \ll N$$

For e encoding of zero at level κ:

$$|e p_{zt} \bmod N| = \left|\sum r_i v_i + a v_0\right| \ll N$$

- **Zero-testing process** : z(e) outputs 1 iff $|e p_{zt} \bmod N| \ll N$

Cryptanalysis

Step 1: "Integer Extraction"

Encoding of zero at level κ :

$$e = \sum r_i u_i + a x_0$$

$$e p_{zt} \bmod N = \sum r_i v_i + a v_0 \qquad \textit{over the integers}$$

«Integer Extraction» :

$$\phi : \sum r_i u_i + a x_0 \mapsto \sum r_i v_i + a v_0$$

- ϕ is well-defined (for r_i's within $]-p_i/2, p_i/2]$).
- $\phi(e) = e p_{zt} \bmod N$ **for small enough e**.
- **For large e**, the key observation is that ϕ is actually $\mathbb{Z}$-linear (for r_i's within $]-p_i/2, p_i/2]$, as above).

Extraction of "Large" Encodings

ϕ can be computed over ladder elements using $\mathbb{Z}$-linearity:

$$\phi(X_0) = X_0 p_{zt} \bmod N$$
$$\phi(X_1) = \phi(X_1 - \alpha X_0) + \alpha\phi(X_0)$$
$$\phi(X_2) = \phi(X_2 - \beta X_1 - \gamma X_0) + \beta\phi(X_1) + \gamma\phi(X_0)$$
$$\cdots$$

Likewise, let *a*, *b* denote two encodings s.t. *ab* is at level κ, then we can compute:

$$\phi(ab) = \phi(ab - \alpha_m X_m - \cdots - \alpha_0 X_0) + \alpha_m\phi(X_m) + \cdots + \alpha_0\phi(X_0)$$

Interlude: Breaking Optimization

ϕ can now be computed for "large" elements.

Optimized scheme: publishes qx_0 for small q to allow smaller ladders.

- **Straightforward application of ϕ:**

$$\begin{aligned} \phi(qx_0) &= qv_0 \\ q &= \gcd(qx_0, qv_0) \\ v_0 &= qv_0/q \\ x_0 &= v_0 p_{zt}^{-1} \bmod N \end{aligned}$$

Step 2: Recovering x_0

Pick : $n+1$ encodings of zero a_i at level 1.

$n+1$ encodings b_i at level $\kappa - 1$.

$$a_i = \mathrm{CRT}_{(p_i)}\left(\frac{a_{i,k}g_i}{z}\right) + a_i' x_0$$

$$b_j = \mathrm{CRT}_{(p_i)}\left(\frac{b_{j,k}}{z^{\kappa-1}}\right) + b_i' x_0$$

We can write :

$$a_i b_j = \sum a_{i,k} b_{j,k} u_k + c_{i,j} x_0$$

$$\phi(a_i b_j) = \sum a_{i,k} b_{j,k} v_k + c_{i,j} v_0$$

Step 2: Recovering x_0

Pick : $n+1$ encodings of zero a_i at level 1.
$n+1$ encodings b_i at level $\kappa-1$.

We have : $$\phi(a_i b_j) = \sum a_{i,k} b_{j,k} v_k + c_{i,j} v_0$$

This is a matrix product **modulo v_0**!

$$\begin{bmatrix} & \vdots & \\ \cdots & \phi(a_i b_j) & \cdots \\ & \vdots & \end{bmatrix} = \begin{bmatrix} & \vdots & \\ \cdots & a_{i,k} & \cdots \\ & \vdots & \end{bmatrix} \begin{bmatrix} \ddots & & 0 \\ & v_k & \\ 0 & & \ddots \end{bmatrix} \begin{bmatrix} & \vdots & \\ \cdots & b_{j,k} & \cdots \\ & \vdots & \end{bmatrix}^T$$

- Rank is ≤ n, so $\det\left(\left[\phi(a_i b_j)\right]\right) = 0 \bmod v_0$.
- $v_0 = \mathrm{pgcd}\left(\det\left(\left[\phi(a_i b_j)\right]\right), \det\left(\left[\phi(a'_i b'_j)\right]\right)\right)$.

Wrapping up the Attack

The attack recovers v_0 in polynomial time.
Then $x_0 = v_0 / p_{zt} \bmod N$.

Knowing x_0 essentialy downgrades CLT15 to CLT13.

All other secret parameters are then recovred as in [CHLRS15].

Bonus: CLT15 gives out free encodings of zero in the form of ladder elements. Makes attack more general than with CLT13.

Conclusion

Bigger Picture

	Key exchange	**Obfuscation**
GGH13	✗	✈ 🚑 war zone
CLT13	✗	✈ 🚑 war zone
GGH15	✗	✗
CLT15	✗	✗

Obfuscation v1 : schemes that use multilinear maps as they are. Multilinear maps have other applications.

Obfuscation v2 : schemes that are aware of existing attacks on multilinear maps. Patch their usage accordingly.

`http://malb.io/are-graded-encoding-schemes-broken-yet.html`

Current Situation

"Generic" multilinear maps are broken (e.g. key exchange). Line of research seems abandoned?

Unresolved issues with obfuscation as noted.

Host of results assuming mmaps are in limbo.

Open problems:

- Further analysis. More clarity is needed.
- Significantly different schemes. Worth noting that mmaps are "too powerful" for some of their applications.

Thank you!