Notes On GGH13 Without Ideals

Martin Albrecht[1] **Alex Davidson**[1] Enrique Larraia[1] Alice Pellet--Mary[2]

[1] Royal Holloway, University of London
[2] ENS de Lyon, Laboratoire LIP (U. Lyon, CNRS, ENSL, INRIA, UCBL), France.

IMACC 2017
Full version: eprint/2017/906

December 13, 2017

Outline of talk

Halevi [eprint/2015/866]:

> *"The core computational hardness problem for GGH13 is to find a representative of the ideal $\langle g \rangle$"*

Outline of talk

Halevi [eprint/2015/866]:

> *"The core computational hardness problem for GGH13 is to find a representative of the ideal $\langle g \rangle$"*

This work:

> *"GGH13-like schemes have structural weaknesses that can be exploited even when the ideal $\langle g \rangle$ is removed"*

Outline of talk

Halevi [eprint/2015/866]:

> *"The core computational hardness problem for GGH13 is to find a representative of the ideal $\langle g \rangle$"*

This work:

> *"GGH13-like schemes have structural weaknesses that can be exploited even when the ideal $\langle g \rangle$ is removed"*

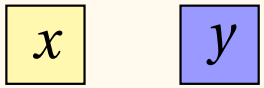
Also:

> *"Can we build better security models for analysing indistinguishability obfuscation candidates?"*

Table of contents

Introduction

Multilinear Maps (MMAPs)

Encodings: 

Multilinear Maps (MMAPs)

Encodings: $\boxed{x}$ $\boxed{y}$

Operations: $\boxed{x} + \boxed{y} = \boxed{x+y}$; $\boxed{x} \cdot \boxed{y} = \boxed{x \cdot y}$

Multilinear Maps (MMAPs)

Encodings: $\boxed{x}$ $\boxed{y}$

Operations: $\boxed{x} + \boxed{y} = \boxed{x+y}$; $\boxed{x} \cdot \boxed{y} = \boxed{x \cdot y}$

Zero-testing: $\mathsf{ZeroTest}(\boxed{x}) = 1 / 0$ if $x = 0 / x \neq 0$ respectively

Multilinear Jigsaw Puzzles (MJPs)

A key component for building indistinguishability obfuscation (IO)

- MJP $\subset$ MMAP
- No low-level encodings of zero

  $\hookrightarrow$ Not vulnerable to 'zeroizing'
- IO does not need low-level zero encodings $\implies$ can use MJPs

We will focus on the GGH13 candidate[1]

[1] Sanjam Garg, Craig Gentry, and Shai Halevi. "Candidate Multilinear Maps from Ideal Lattices". In: 2013, pp. 1–17. DOI: 10.1007/978-3-642-38348-9_1.

GGH13 scheme

All encodings live in a ring $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$ for some ring $\mathcal{R}$

Plaintexts are sampled from $\mathcal{R}_g$ for some 'small' $g \in \mathcal{R}$

GGH13 scheme

All encodings live in a ring $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$ for some ring $\mathcal{R}$

Plaintexts are sampled from $\mathcal{R}_g$ for some 'small' $g \in \mathcal{R}$

Encoding of α at level ℓ:

$$[\alpha]_\ell = (\alpha + rg)/z_\ell \pmod{q}$$

- $r \leftarrow\!\!\$\ \chi$ (χ is a distribution over 'small' elements in $\mathcal{R}_q$)
- $z_\ell \leftarrow\!\!\$\ \mathcal{R}_q$

GGH13 scheme

All encodings live in a ring $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$ for some ring $\mathcal{R}$

Plaintexts are sampled from $\mathcal{R}_g$ for some 'small' $g \in \mathcal{R}$

Encoding of α at level ℓ:

$$[\alpha]_\ell = (\alpha + rg)/z_\ell \pmod{q}$$

- $r \leftarrow_\$ \chi$ (χ is a distribution over 'small' elements in $\mathcal{R}_q$)
- $z_\ell \leftarrow_\$ \mathcal{R}_q$

Addition: $[\alpha_0]_\ell + [\alpha_1]_\ell = [\alpha_0 + \alpha_1]_\ell$

Multiplication: $[\alpha_0]_{\ell_0} \cdot [\alpha_1]_{\ell_1} = [\alpha_0\alpha_1]_{\ell_0\ell_1}$

Zero-testing

Total multilinearity: $\kappa \in \mathbb{N}$

Parameter: $\mathsf{ztk} = (h \prod_{\ell=1}^{\kappa} z_\ell)/g$ (h is 'small')

Zero-testing

Total multilinearity: $\kappa \in \mathbb{N}$

Parameter: $\mathsf{ztk} = (h \prod_{\ell=1}^{\kappa} z_\ell)/g$ (h is 'small')

Zero-test:

$$\delta = \mathsf{ztk} \cdot [\alpha]_\kappa = h(\alpha/g + r)$$

Zero-testing

Total multilinearity: $\kappa \in \mathbb{N}$

Parameter: $\mathsf{ztk} = (h \prod_{\ell=1}^{\kappa} z_\ell)/g$ (h is 'small')

Zero-test:

$$\delta = \mathsf{ztk} \cdot [\alpha]_\kappa = h(\alpha/g + r)$$

$\hookrightarrow$ if $\alpha = 0$ *then* $\delta = hr \implies$ 'small'

$\hookrightarrow$ *else* δ is uniform in $\mathcal{R}_q$

Vulnerabilities

Numerous attacks on GGH13 scheme:

$\hookrightarrow$ Using low-level zero encodings $\Longrightarrow$ 'zeroizing'[2] *

$\hookrightarrow$ Using top-level zero encodings $\Longrightarrow$ 'annihilation'[3] *, **

$\hookrightarrow$ No zero encodings $\Longrightarrow$ 'subfield lattice'[4]

* acquire knowledge of $\langle g \rangle$

** work against obfuscation candidates

---

[2] Yupu Hu and Huiwen Jia. "Cryptanalysis of GGH Map". In: 2016, pp. 537–565. DOI: 10.1007/978-3-662-49890-3_21.

[3] Eric Miles, Amit Sahai, and Mark Zhandry. "Annihilation Attacks for Multilinear Maps: Cryptanalysis of Indistinguishability Obfuscation over GGH13". In: 2016, pp. 629–658. DOI: 10.1007/978-3-662-53008-5_22.

[4] Martin R. Albrecht, Shi Bai, and Léo Ducas. "A Subfield Lattice Attack on Overstretched NTRU Assumptions - Cryptanalysis of Some FHE and Graded Encoding Schemes". In: 2016, pp. 153–178. DOI: 10.1007/978-3-662-53018-4_6, Jung Hee Cheon, Jinhyuck Jeong, and Changmin Lee. *An Algorithm for NTRU Problems and Cryptanalysis of the GGH Multilinear Map without a Low Level Encoding of Zero.* Cryptology ePrint Archive, Report 2016/139. http://eprint.iacr.org/2016/139. 2016.

Annihilation attack

$$\boxed{0} = (\gamma_1 g + \gamma_2 g^2 + \ldots + \gamma_\kappa g^\kappa)/z_\kappa$$

$$\delta_i = \mathsf{ZeroTest}(\boxed{0}) = \gamma_{1,i} + \gamma_{2,i}\, g + \ldots + \gamma_{\kappa,i}\, g^{\kappa-1}$$

[5] Eric Miles, Amit Sahai, and Mark Zhandry. "Annihilation Attacks for Multilinear Maps: Cryptanalysis of Indistinguishability Obfuscation over GGH13". In: 2016, pp. 629–658. DOI: 10.1007/978-3-662-53008-5_22.

Annihilation attack

$$\boxed{0} \;=\; (\gamma_1 g + \gamma_2 g^2 + \ldots + \gamma_\kappa g^\kappa)/z_\kappa$$

$$\delta_i \;=\; \mathsf{ZeroTest}(\;\boxed{0}\;) \;=\; \gamma_{1,i} + \gamma_{2,i}\, g + \ldots + \gamma_{\kappa,i}\, g^{\kappa-1}$$

$\gamma_{j,i}$ is a polynomial in $\{\alpha\}, \{r\}$ values

$\hookrightarrow$ degree j in $\{r\}$ (*unknown*) and $(\kappa - j)$ in $\{\alpha\}$ (*known*)

[5] Eric Miles, Amit Sahai, and Mark Zhandry. "Annihilation Attacks for Multilinear Maps: Cryptanalysis of Indistinguishability Obfuscation over GGH13". In: 2016, pp. 629–658. DOI: 10.1007/978-3-662-53008-5_22.

Annihilation attack

$$\boxed{0} \;=\; (\gamma_1 g + \gamma_2 g^2 + \ldots + \gamma_\kappa g^\kappa)/z_\kappa$$

$$\delta_i \;=\; \mathsf{ZeroTest}(\;\boxed{0}\;) \;=\; \gamma_{1,i} + \gamma_{2,i}\, g + \ldots + \gamma_{\kappa,i}\, g^{\kappa-1}$$

$\gamma_{j,i}$ is a polynomial in $\{\alpha\}, \{r\}$ values

$\hookrightarrow$ degree j in $\{r\}$ (*unknown*) and $(\kappa - j)$ in $\{\alpha\}$ (*known*)

Find 'annihilating' polynomial[5] P such that $P(\{\gamma_{1,i}\}_i) = 0$

$\hookrightarrow$ then $P(\{\delta_i\}_i) \in \langle g \rangle \implies$ can form basis of $\langle g \rangle$ (heuristically)

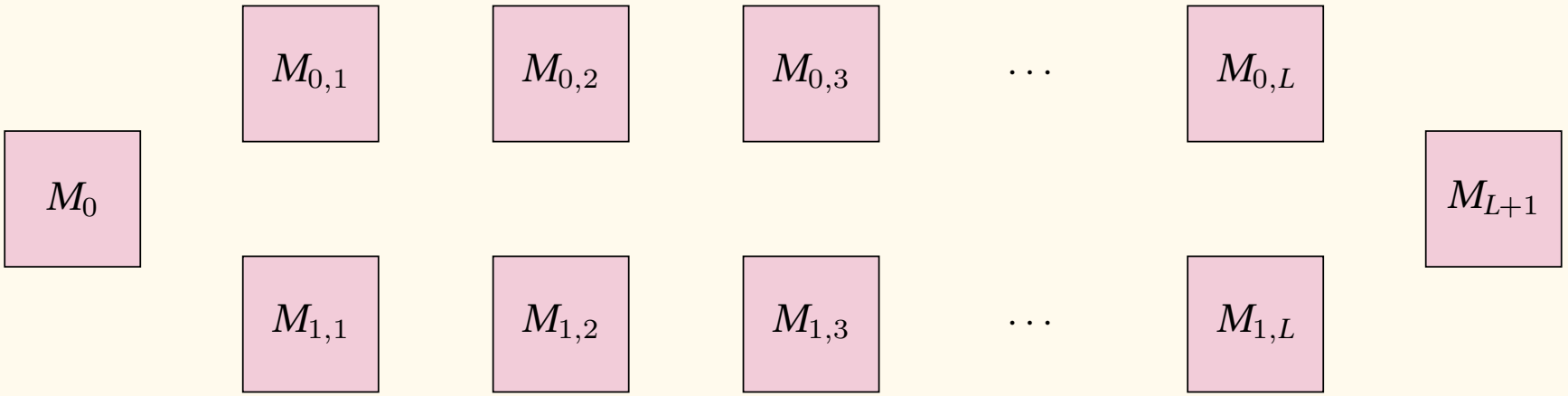
$\hookrightarrow$ use $\langle g \rangle$ in IO security model to distinguish obfuscated circuits

[5] Eric Miles, Amit Sahai, and Mark Zhandry. "Annihilation Attacks for Multilinear Maps: Cryptanalysis of Indistinguishability Obfuscation over GGH13". In: 2016, pp. 629–658. DOI: 10.1007/978-3-662-53008-5_22.

BGK-style obfuscators

Matrix branching programs

$C(x) \in \{0,1\} \to \mathcal{M}(x) \in \mathcal{R}_q$ (Barrington's theorem) :

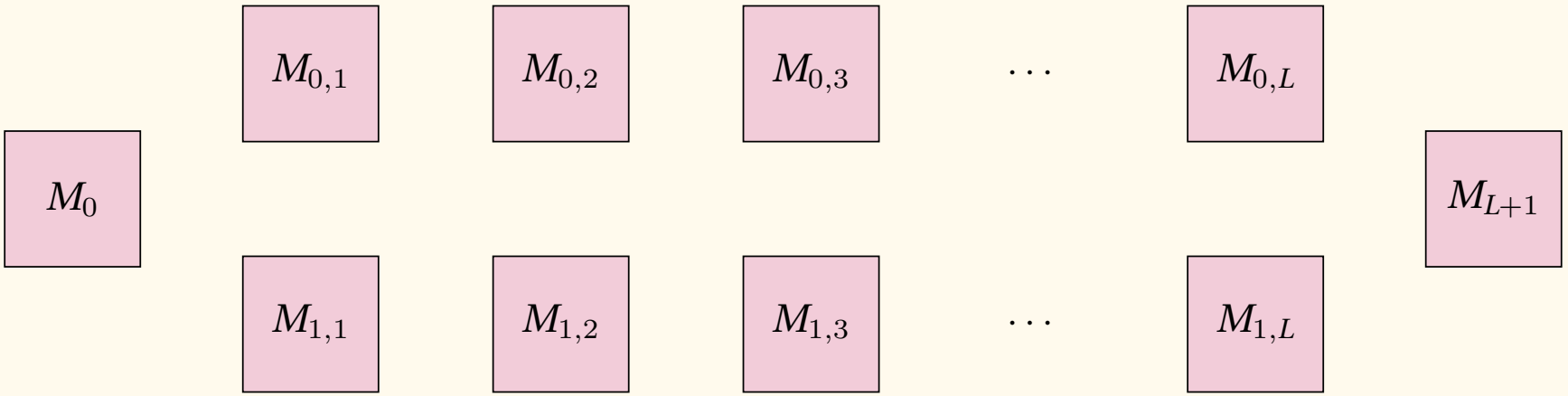


$\mathsf{inp} : [L] \mapsto [\nu]$ is an input function, $x \in \{0,1\}^\nu$ is an input

$\hookrightarrow M_0, M_{L+1} \in \mathcal{R}^5$, $M_{b,l} \in \mathcal{R}^{5\times 5}$

Matrix branching programs

$C(x) \in \{0,1\} \to \mathcal{M}(x) \in \mathcal{R}_q$ (Barrington's theorem) :



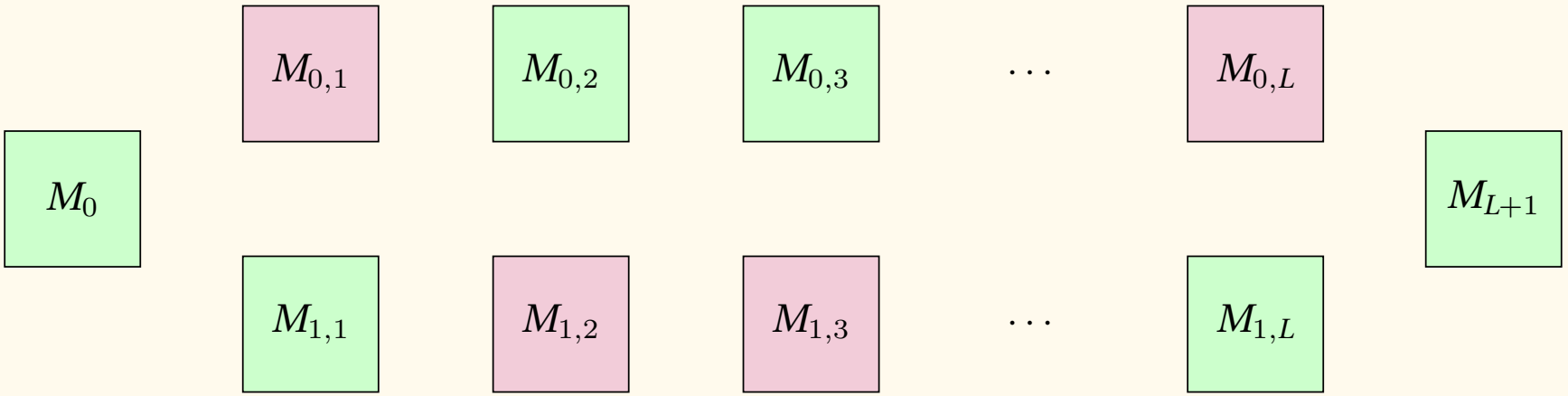
$\mathsf{inp} : [L] \mapsto [\nu]$ is an input function, $x \in \{0,1\}^\nu$ is an input

$\hookrightarrow M_0, M_{L+1} \in \mathcal{R}^5$, $M_{b,l} \in \mathcal{R}^{5\times 5}$

$\hookrightarrow C(x) = \mathcal{M}(x) = M_0 \left(\prod_{i=1} M_{x_{\mathsf{inp}(i),i}}\right) M_{L+1} \in \{0, \neq 0\}$

Matrix branching programs

$C(x) \in \{0,1\} \to \mathcal{M}(x) \in \mathcal{R}_q$ (Barrington's theorem) :



$\mathsf{inp} : [L] \mapsto [\nu]$ is an input function, $x \in \{0,1\}^\nu$ is an input

$\hookrightarrow M_0, M_{L+1} \in \mathcal{R}^5, M_{b,l} \in \mathcal{R}^{5\times 5}$

$\hookrightarrow C(x) = \mathcal{M}(x) = M_0 \left(\prod_{i=1} M_{x_{\mathsf{inp}(i),i}}\right) M_{L+1} \in \{0, \neq 0\}$

Example: $\mathsf{inp}(i) = i \pmod \nu$, $L = 0 \pmod \nu$, $x = 100\ldots1$

BGK-style obfuscation (I)

Kilian randomisation

$$\widehat{M_{b,l}} = R_{l-1}^{-1} \cdot M_{b,l} \cdot R_l$$

$$\widehat{M_0} = M_0 \cdot R_0$$

$$\widehat{M_{L+1}} = R_L^{-1} \cdot M_{L+1}$$

BGK-style obfuscation (I)

Kilian randomisation

$$\widehat{M_{b,l}} = R_{l-1}^{-1} \cdot M_{b,l} \cdot R_l$$

$$\widehat{M_0} = M_0 \cdot R_0$$

$$\widehat{M_{L+1}} = R_L^{-1} \cdot M_{L+1}$$

Input-mixing scalars

$$\widehat{M_{b,l}}' = \epsilon_{b,l} \widehat{M_{b,l}}$$

$$\widehat{M_0}' = \epsilon_0 \widehat{M_0}$$

$$\widehat{M_{L+1}}' = \epsilon_{L+1} \widehat{M_{L+1}}$$

BGK-style obfuscation (II)

Encoding

$$\widetilde{M_0} = \left[\widehat{M_0}'\right]_0, \qquad \widetilde{M_{L+1}} = \left[\widehat{M_{L+1}}'\right]_{L+1}$$

$$\widetilde{M_{b,l}} = \left[\widehat{M_{b,l}}'\right]_l$$

BGK-style obfuscation (II)

Encoding

$$\widetilde{M_0} = \left[\widehat{M_0}'\right]_0, \qquad \widetilde{M_{L+1}} = \left[\widehat{M_{L+1}}'\right]_{L+1}$$

$$\widetilde{M_{b,l}} = \left[\widehat{M_{b,l}}'\right]_l$$

Evaluation

$$[\tilde{\epsilon}\alpha]_{\mathcal{U}} = \widetilde{M_0} \cdot \prod_{l=1}^{L} \widetilde{M_{x_{\mathsf{inp}(l)},l}} \cdot \widetilde{M_{L+1}}$$

BGK-style obfuscation (II)

Encoding

$$\widetilde{M_0} = \left[\widehat{M_0}'\right]_0, \quad \widetilde{M_{L+1}} = \left[\widehat{M_{L+1}}'\right]_{L+1}$$

$$\widetilde{M_{b,l}} = \left[\widehat{M_{b,l}}'\right]_l$$

Evaluation

$$[\tilde{\epsilon}\alpha]_{\mathcal{U}} = \widetilde{M_0} \cdot \prod_{l=1}^{L} \widetilde{M_{x_{\mathsf{inp}(l)},l}} \cdot \widetilde{M_{L+1}}$$

If $C(x) = 0$, then $[\tilde{\epsilon}\alpha]_{\mathcal{U}}$ is an encoding of zero

$\hookrightarrow$ use zero-test to learn output of obfuscated circuit

GGH13 without ideals

Encodings and operations

Intuition: GGH13 in MSB

Encoding at level ℓ:

$$[\alpha]_\ell = (\alpha + r/\beta_\ell)/z_\ell \pmod{q}$$

- $\beta_\ell \leftarrow \mathcal{R}_q$ such that $\|\beta_\ell\|_\infty \leq \sqrt[\kappa]{q}$ (using canonical embeddings)
- $[\alpha_0]_\ell + [\alpha_1]_\ell = [\alpha_0 + \alpha_1]_\ell$
- $[\alpha_0]_{\ell_0} \cdot [\alpha_1]_{\ell_1} = [\alpha_0 \alpha_1]_{\ell_0 \ell_1}$

Zero-testing

Parameter: $\mathsf{ztk} = \prod_{\ell=1}^{\kappa} \beta_\ell \cdot z_\ell$

$\mathsf{ZeroTest}(\mathsf{ztk}, [\alpha]_\kappa)$:

$$\mathsf{ztk} \cdot [\alpha]_\kappa = \delta = \prod_{\ell=1}^{\kappa} \beta_\ell \cdot \alpha + \beta^{(\kappa-1)}\gamma_1 + \ldots + \beta^{(1)}\gamma_{\kappa-1} + \gamma_\kappa \bmod q$$

$\hookrightarrow \gamma_i = \mathsf{poly}(\{\alpha_i\}_i, \{r_i\}_i)$, $\beta^{(j)} =$ 'sum of degree j monomials in $\{\beta_\ell\}$'

Zero-testing

Parameter: $\mathsf{ztk} = \prod_{\ell=1}^{\kappa} \beta_\ell \cdot z_\ell$

$\mathsf{ZeroTest}(\mathsf{ztk}, [\alpha]_\kappa)$:

$$\mathsf{ztk} \cdot [\alpha]_\kappa = \delta = \prod_{\ell=1}^{\kappa} \beta_\ell \cdot \alpha + \beta^{(\kappa-1)}\gamma_1 + \ldots + \beta^{(1)}\gamma_{\kappa-1} + \gamma_\kappa \bmod q$$

$\hookrightarrow \gamma_i = \mathsf{poly}(\{\alpha_i\}_i, \{r_i\}_i)$, $\beta^{(j)} =$ 'sum of degree j monomials in $\{\beta_\ell\}$'

$\hookrightarrow \alpha = 0 \implies \|\delta\|_\infty \leq q$; $\alpha \neq 0 \implies q \leq \|\delta\|_\infty$

Zero-testing

Parameter: $\mathsf{ztk} = \prod_{\ell=1}^{\kappa} \beta_\ell \cdot z_\ell$

$\mathsf{ZeroTest}(\mathsf{ztk}, [\alpha]_\kappa)$:

$$\mathsf{ztk} \cdot [\alpha]_\kappa = \delta = \prod_{\ell=1}^{\kappa} \beta_\ell \cdot \alpha + \beta^{(\kappa-1)}\gamma_1 + \ldots + \beta^{(1)}\gamma_{\kappa-1} + \gamma_\kappa \bmod q$$

$\hookrightarrow \gamma_i = \mathsf{poly}(\{\alpha_i\}_i, \{r_i\}_i)$, $\beta^{(j)} =$ 'sum of degree j monomials in $\{\beta_\ell\}$'

$\hookrightarrow \alpha = 0 \implies \|\delta\|_\infty \leq q$; $\alpha \neq 0 \implies q \leq \|\delta\|_\infty$

$\hookrightarrow$ **No longer any ideals to use in attacks**

Security models

Current IO security models

Ideal/weakened graded encoding models

$\hookrightarrow$ CGH17[6] suggests that weakened GEM is insufficient

$\hookrightarrow$ e.g. GMMSSZ16[7] secure using unmodelled characteristics

[6] Yilei Chen, Craig Gentry, and Shai Halevi. "Cryptanalyses of Candidate Branching Program Obfuscators". In: 2017, pp. 278–307.

[7] Sanjam Garg et al. "Secure Obfuscation in a Weak Multilinear Map Model". In: 2016, pp. 241–268. DOI: 10.1007/978-3-662-53644-5_10.

Current IO security models

Ideal/weakened graded encoding models

$\hookrightarrow$ CGH17[6] suggests that weakened GEM is insufficient

$\hookrightarrow$ e.g. GMMSSZ16[7] secure using unmodelled characteristics

We use game-based models **IND-$\mathcal{M}$** and **IND-OBF** for attacks

Adversary has oracle access to actual MJP

IND-OBF is sufficient for GGH13 attack

---

[6] Yilei Chen, Craig Gentry, and Shai Halevi. "Cryptanalyses of Candidate Branching Program Obfuscators". In: 2017, pp. 278–307.

[7] Sanjam Garg et al. "Secure Obfuscation in a Weak Multilinear Map Model". In: 2016, pp. 241–268. DOI: 10.1007/978-3-662-53644-5_10.

Indistinguishability (I)

Branching programs (**IND-$\mathcal{M}$**)

Game **IND-$\mathcal{M}^{\mathcal{A}}(\lambda)$**:

1. $(\mathsf{sp}, \mathsf{prms}, \mathsf{ztk}) \leftarrow \mathsf{JInstGen}(1^{\lambda}, 1^{\kappa})$
2. $(\mathsf{st}, \mathcal{M}_0, \mathcal{M}_1) \leftarrow \mathcal{A}_0(\kappa, \mathsf{prms})$
3. $b \leftarrow \{0,1\}$
4. $\widehat{\mathcal{M}}_b \leftarrow \mathsf{Encode}(\mathsf{sp}, \mathsf{prms}, \{\mathcal{S}_i\}_{i\in[\kappa]}, \mathcal{M}_b)$
5. $b' \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{zt}}}(\mathsf{st})$
6. output $(b' = b)$

Oracle $\mathcal{O}_{\mathsf{zt}}(x)$:

1. **if** init, $q \leftarrow 0$; **else**, $q \leftarrow q + 1$
2. **if** $q > \mathcal{Q}$, $\delta \leftarrow \perp$
3. **else**:
4. $\quad [\mu_x]_{\mathcal{U}} \leftarrow \widehat{\mathcal{M}}_b(x)$
5. $\quad \delta_x \leftarrow \mathsf{ZeroTest}(\mathsf{ztk}, [\mu_x]_{\mathcal{U}})$
6. return δ_x

Indistinguishability (II)

Obfuscated branching programs (**IND-OBF**)

Game $\mathbf{IND\text{-}OBF}^{\mathcal{A}}(\lambda)$:

1. $(\mathsf{sk}, \mathsf{prms}, \mathsf{ztk}) \leftarrow \mathsf{JInstGen}(1^\lambda, 1^\kappa)$
2. $(\mathsf{st}, \mathcal{M}_0, \mathcal{M}_1) \leftarrow \mathcal{A}_0(\kappa, \mathsf{prms})$
3. $b \leftarrow \{0,1\}$
4. $\widehat{\mathcal{M}}_b \leftarrow \mathsf{iO}(\mathsf{sp}, \mathsf{prms}, \{\mathcal{S}_i\}_{i\in[\kappa]}, \mathcal{M}_b)$ <<----
5. $b' \leftarrow \mathcal{A}_1^{\mathcal{O}_{\mathsf{zt}}}(\mathsf{st})$
6. output $(b' = b)$

We can embed any IO candidate as the oracle iO

$\hookrightarrow$ analyse security when this is a BGK-style obfuscator

Attacks

Attack in **IND-$\mathcal{M}$**

Encoding of zero:

$$\delta_x = \beta^{(\kappa-1)}\gamma_{1,x} + \ldots + \beta^{(1)}\gamma_{\kappa-1,x} + \gamma_{\kappa,x} \mod q$$

- $\gamma_{1,x}$ is almost the same as before (linear in unknown $\{r_i\}_i$)
- Stratified into κ different monomials of degree $\kappa - 1$ in β_ℓ
- We aim to annihilate this term in a similar way
- Notice: β_ℓ terms can be ignored

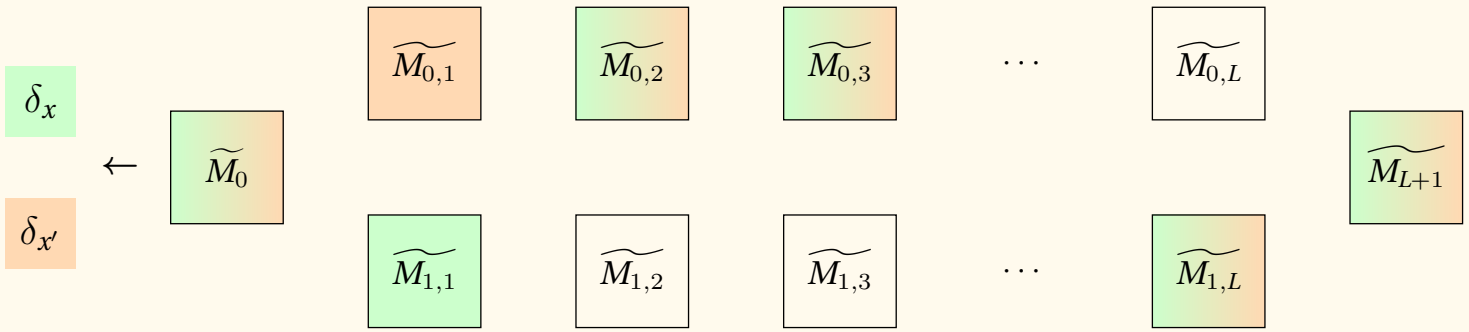
Attack in **IND-$\mathcal{M}$**

Use $\mathcal{M}_0, \mathcal{M}_1$ where $\mathcal{M}_b(x) = 0$ for large number of inputs

Attack in **IND-$\mathcal{M}$**

Use $\mathcal{M}_0, \mathcal{M}_1$ where $\mathcal{M}_b(x) = 0$ for large number of inputs

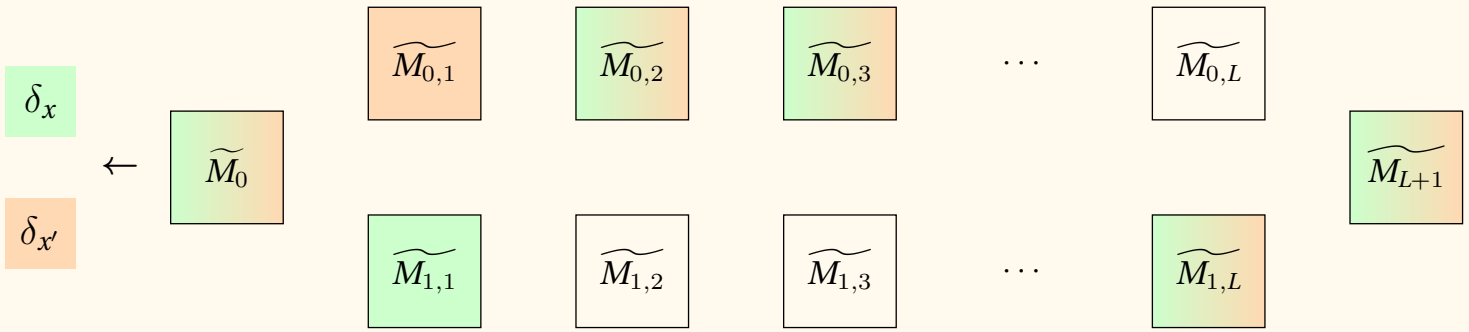
Evaluate (two inputs x, x'):



Attack in **IND-$\mathcal{M}$**

Use $\mathcal{M}_0, \mathcal{M}_1$ where $\mathcal{M}_b(x) = 0$ for large number of inputs

Evaluate (two inputs x, x'):



Let $r_{i,j,l}$ denote the (i, j) entry in the matrix at level $l \neq 1$

$c_{i,j,l}^{x_{\mathsf{inp}(l)}}$, $c_{i,j,l}^{x'_{\mathsf{inp}(l)}}$ are the *known* coefficients of each $r_{i,j,l}$

Attack in **IND-$\mathcal{M}$**

Computing

$$\delta' = c_{i,j,l}^{x'_{\mathsf{inp}(l)}} \delta_x - c_{i,j,l}^{x_{\mathsf{inp}(l)}} \delta_{x'} \pmod{q}$$

removes $r_{i,j,l}$ for any i, j, l

Attack in **IND-$\mathcal{M}$**

Computing

$$\delta' = c_{i,j,l}^{x'_{\mathsf{inp}(l)}} \delta_x - c_{i,j,l}^{x_{\mathsf{inp}(l)}} \delta_{x'} \ (\bmod \ q)$$

removes $r_{i,j,l}$ for any i, j, l

$\hookrightarrow$ total of $(25(\kappa + 1) + 10)$ variables

Attack in **IND-$\mathcal{M}$**

Computing

$$\delta' = c_{i,j,l}^{x'_{\mathsf{inp}(l)}} \delta_x - c_{i,j,l}^{x_{\mathsf{inp}(l)}} \delta_{x'} \ (\bmod \ q)$$

removes $r_{i,j,l}$ for any i, j, l

$\hookrightarrow$ total of $(25(\kappa + 1) + 10)$ variables

$\hookrightarrow$ recompute δ' until all $r_{i,j,l}$ are eliminated using 4 inputs

Attack in **IND-$\mathcal{M}$**

Computing

$$\delta' = c_{i,j,l}^{x'_{\mathsf{inp}(l)}} \delta_x - c_{i,j,l}^{x_{\mathsf{inp}(l)}} \delta_{x'} \ (\bmod \ q)$$

removes $r_{i,j,l}$ for any i, j, l

$\hookrightarrow$ total of $(25(\kappa+1)+10)$ variables

$\hookrightarrow$ recompute δ' until all $r_{i,j,l}$ are eliminated using 4 inputs

After: $\delta' = \beta^{(\kappa-2)}\gamma'_{2,x} + \ldots + \beta^{(1)}\gamma'_{\kappa-1,x} + \gamma'_{\kappa,x} \ (\bmod \ q)$

$\hookrightarrow$ noticeably smaller than δ_{x^*} for any input x^*

$$\|\delta_{x^*}\|_\infty \geq q^{(\kappa-2)/\kappa} \geq \|\delta'\|_\infty$$

Distinguishes $\mathcal{M}_0, \mathcal{M}_1$ for equivalent functionalities

Attack in **IND-OBF**

In **IND-OBF** we introduce the input-mixing scalars

$\hookrightarrow$ Appear to thwart the previous attack

$\hookrightarrow$ Each input introduces a new scalar

Attack in **IND-OBF**

In **IND-OBF** we introduce the input-mixing scalars

$\hookrightarrow$ Appear to thwart the previous attack

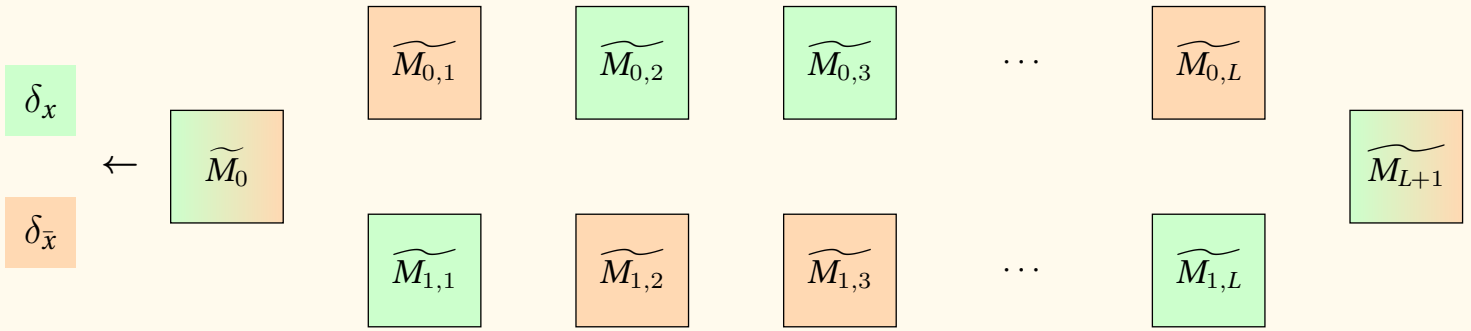
$\hookrightarrow$ Each input introduces a new scalar

However the attack can be adapted...

$\hookrightarrow$ Use multiple inputs that include all scalars

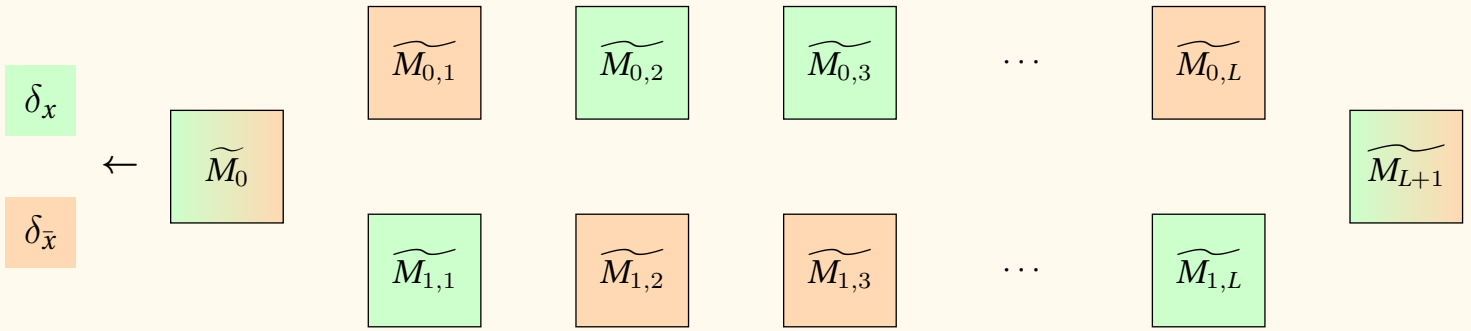
Attack in **IND-OBF**

Evaluate inputs x and $\bar{x} = x \oplus 1^{\nu}$:



Attack in **IND-OBF**

Evaluate inputs x and $\bar{x} = x \oplus 1^{\nu}$:



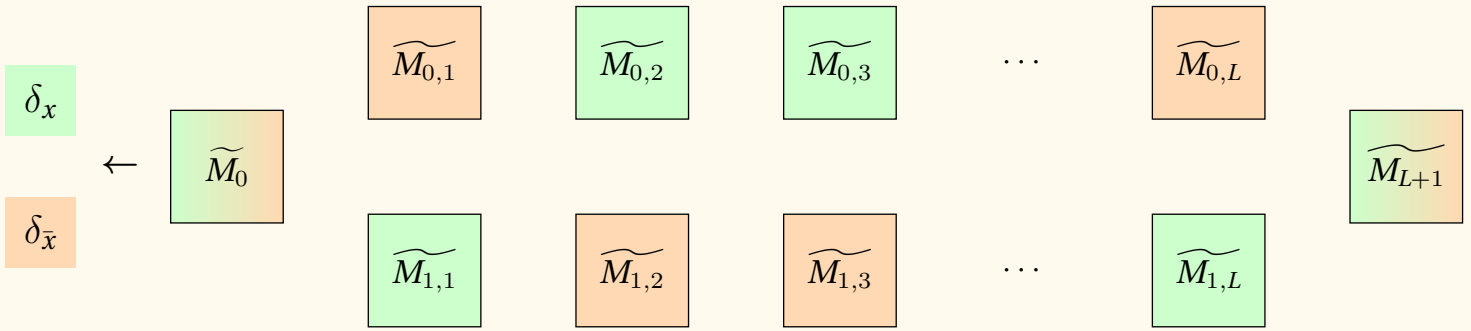
- Annihilate largest component of $\delta^* = \delta_x \cdot \delta_{\bar{x}}$

  $\hookrightarrow$ quadratic in $\beta^{\kappa-1}$ terms

  $\hookrightarrow \epsilon^* = (\epsilon_0 \epsilon_\kappa)^2 \prod_{i=1}^{\kappa-1} \epsilon_{0,i} \cdot \epsilon_{1,i}$

Attack in **IND-OBF**

Evaluate inputs x and $\bar{x} = x \oplus 1^{\nu}$:



- Annihilate largest component of $\delta^* = \delta_x \cdot \delta_{\bar{x}}$

  $\hookrightarrow$ quadratic in $\beta^{\kappa-1}$ terms

  $\hookrightarrow \epsilon^* = (\epsilon_0 \epsilon_\kappa)^2 \prod_{i=1}^{\kappa-1} \epsilon_{0,i} \cdot \epsilon_{1,i}$

- $\delta^* = \delta_x \cdot \delta_{\bar{x}} = \epsilon^* \cdot p_x(\{r_{i,j,l}, r_{i',j',l'}\}) +$ smaller terms $(\bmod q)$

  $\hookrightarrow p_x$ is a linear function of $(50(\kappa-1)+10)^2$ unknowns

  $\hookrightarrow$ Attack proceeds as in **IND-$\mathcal{M}$**

Concluding remarks

Structural faults in GGH13

- Appears algebraic structure of GGH13 encodings is weak
- Unrelated to the presence of ideals
- Introduce similar encoding scheme

  $\hookrightarrow$ Zero-testing explicitly based on magnitude
- Scheme is vulnerable to adaptations of previous attacks
- Any cases where two schemes differ?

Security models

- **IND-$\mathcal{M}$** and **IND-OBF** are sufficient game-based models
- Allow adversary greater access to MJP scheme
- Seems preferable to use instead of weakened GEM

  $\hookrightarrow$ Attacks such as CGH17[8] exploit unmodelled characteristics[9]

  $\hookrightarrow$ e.g. single-input vs dual-input obfuscation

[8] Yilei Chen, Craig Gentry, and Shai Halevi. "Cryptanalyses of Candidate Branching Program Obfuscators". In: 2017, pp. 278–307.

[9] Sanjam Garg et al. "Secure Obfuscation in a Weak Multilinear Map Model". In: 2016, pp. 241–268. DOI: 10.1007/978-3-662-53644-5_10.

Thanks for listening

Questions?