Recent Advances on Foundations of Program Obfuscation

Huijia (Rachel) Lin
UW
Recent Advances
Based on works [Agr18] [AJS18] [LM18]
+ follow-ups [JLMS19] [JLS19]

Shweta Agarwal
Prabhanjan Ananth
Aayush Jain
Christian Matt
Amit Sahai
Program Obfuscation

Simple and Weak Pseudo Randomness Generators
+ LWE + Bilinear Maps
Program Obfuscation

Goal: *Efficiently* transform a program into one that *is functionally equivalent & unintelligible*

\[
\begin{array}{ccc}
\Pi & \xrightarrow{\text{Obfuscator}} & \bar{\Pi} \\
\downarrow & & \downarrow \\
\text{Main code} & \xrightarrow{\text{Efficiently transform}} & \text{Obfuscated code} \\
\end{array}
\]

Mildly Blow-up in Size
Indistinguishability Obfuscation (IO) [BGIRSVDY01]

Hide implementation difference
No general impossibility
Classical Crypto

- Short signature
- Trapdoor permutation
- Public key encryption
- Identity-Based Encryption
- Attribute-Based Encryption
- Fully Homomorphic Encryption
- Multiparty Computation
- (Non-Interactive) Zero-Knowledge

Powerful Abstraction

- Correlation Intractable Hash from LWE
- Two-Round MPC from 2-rnd OT

New Crypto

- Functional Encryption
- Witness Encryption
- (Doubly) Deniable Encryption
- Hardness of Finding Nash
- Correlation Intractable Hash
- Secret Sharing for NP
- Multi-Party Non-Interactive Key Exchange
- OWF with poly hard core bits
- Succinct Garbled RAM
- Two-Round MPC
- Constant Round Concurrent ZK

10 + minimal crypto (e.g., OWFs)
IO?

IO from M-linear maps
• Ultimate Prize: IO from Bilinear Maps
• Minimizing the degree M

IO for limited class of functions from standard assumptions
e.g., VBB for compute-\&-compare from LWE
[WZ17,GKW17]

IO from new math
e.g. tensor products
[GJ18]
First Generation IO

[GGHRSW13, BR14, BGKPS14, PST14, GLSW14, AGIS14, Zim15, AB15, GMMSSZ16, DGGMM16]

M-linear map for poly M

[BS02, Rot13, GGH13]

IO resisting Zeroizing Attacks

[GMMSSZ16, DGGMM16, CVW18]

Candidates

[GGH13, CLT13, GGH15, CLT15]

Zeroizing Attacks

[GGH13, CHL15, GGH15, CLT15]

Direct Attacks

[MSZ16, ADGM17, CGH17]

More Attacks

[Pellet-Mary18, CHK18]
Other types of “simple” PRG?

Second Generation IO

[AJ15, BV15, LPST16, LPST16b, L16, LV16, AS17, L17, LT17]

3-linear map
+ “simple” PRG, block-locality 3
+ LWE

“super simple” PRG, block Locality 2, Impossible * [LV17, BBKK17]

*except for tiny expansion window \( n2^b(1+\varepsilon) \)
∆RG: Perturbation Resilient Generators [AJS18]

PFG: Pseudo Flawed-smudging Generators [LM18]

Third Generation IO
[Agr18, AJS18, LM18, JLMS19, JLS19]

2-linear map
+ “simple” and “weak” PRG
+ LWE
\[ \Delta \text{RG: Perturbation Resilient Generators [AJS18]} \]

\[ \text{PFG: Pseudo Flawed-smudging Generators [LM18]} \]

**Flawed Intuition**

“Simple” Deg 2 poly in \( Z_p \) with **small outputs**

(small = poly, large = super-poly)

“Weak” pseudo-randomness

Outputs smudge/hide even **smaller** LWE noises [also in Agr18]

\[ s \begin{bmatrix} d \end{bmatrix} \leftarrow S \text{ in } Z_p^n \]

\[ r \in Z_p^m, m = n^{1+\epsilon} \]

\[ r_i \]

\[ r_i + e_i \]

\[ \approx \]
SK for $NC^0$

single $f$, long-output, $\in NC^0$

$sk(f)$

Dec

$\{y = f(x)\}_x$

$\{ct(x)\}_x$

• Correctness: Reveal $\{y\}_x$

• Privacy: Reveal only $\{y\}_x$

• Compact: $|ct(x)| = poly(|x|)|f|^{1-\epsilon}$

Starting Point

$[AJ15, BV15, AJS15, LPST16, BNPW17, KNT18]$

+ Block-local PRG

$[L16, LV16, L17, AS17, LT17]$

1-key, compact, $NC^0$

PFG

Special HE

Partially Hiding FE

Secret-Sharing VBB
FE for NC$^0$  vs  Homomorphic Encryption

- Privacy: Reveal only $y$  

- Privacy: Reveal nothing

Need to decrypt, privately
Bootstrapping via HE

FE for NC⁰  Simple \( \overline{\text{FE}} \)  HE

Ideally deg 2 FE [BCFG17, Lin17]

- Privacy: Reveal only \( y \)
- Simplicity: \( f > h \)

\[
\begin{align*}
sk(f) & \quad \overline{sk}(h?) \\
ct(x) & \quad \overline{ct}(?, ?) \\
hct_s(x) & \quad hct_f(y)
\end{align*}
\]

Basic Approach
[GVW12, GKP+13, GVW15, AR17, Agr18]

Use FE to decrypt
FE for NC^0 Simple \overline{FE}

\[ sk(f) \quad \overline{sk}(h) \]

\[ ct(x) \quad \overline{ct}(hct, s) \quad hct_S(x) \]

• Privacy: Reveal only \( y \)

\[ h(hct, s): \]
1. \( hct_f \leftarrow h\text{Eval}(f, hct) \)
2. \( y = h\text{Dec}(s, hct_f) = \overline{L\text{Dec}}(s, hct_f) \mod 2 \)

• Simplicity: \( f > h \)
\( h\text{Dec} \in \text{NC}^1 \)
\( h\text{Eval} > f \)

First Attempt
[GVW12]
FE for NC\(^0\)  Simple \(\overline{\text{FE}}\)

\[
\begin{align*}
    sk(f) &\quad \overline{sk}(h) \\
    ct(x) &\quad \overline{ct}(hct, s) & hct_s(x)
\end{align*}
\]

\begin{itemize}
    \item Privacy: Reveal only key \(ssk(f, s, x)\)
    \item Simplicity: \(f > h\)
\end{itemize}

\text{Half Decrypt for Simplicity} [GVW15]

\[
h(hct, s):\]
\[
    \begin{align*}
        1. \quad & hct_f \leftarrow \text{hEval}(f, hct) \\
        2. \quad & y + 2e = \text{LDec}(s, hct_f)
    \end{align*}
\]

\(\text{LDec} \in \deg 2\)

\(\text{hEval} > f\)
FE for NC⁰  Simple \( \overline{\text{FE}} \)

\[
\begin{align*}
\text{sk}(f) & \quad \overline{\text{sk}}(h) \\
c\text{t}(x) & \quad \overline{\text{c}}\text{t}(hct, s, sd) \quad hct_s(x)
\end{align*}
\]

- Privacy: \( \text{Reveal FGF}(sd)\) hides \( f \) \( s \) \( x \)
- Simplicity: \( f > h \)
  \( \text{LDec} \in \text{deg} \ 2 \)
  \( \text{hEval} > f \)

\( h(hct, s, sd) \):
1. \( hct_f \leftarrow \text{hEval}(f, hct) \)
2. \( y + 2e = \text{LDec}(s, hct_f) \)
3. \( y + 2e + 2\text{PFG}(sd) \)

PFG for Privacy

[ajs18, agr18, lm18]
FE for NC\(^0\)  Simple \(\overline{\text{FE}}\)

\[
\begin{align*}
sk(f) & & \overline{sk}(h) \\
ct(x) & & \overline{ct}(hct, s, sd) \ hct_{s}(x)
\end{align*}
\]

- Privacy: \(e + \text{PFG}(sd)\) hides \(e\)
- Simplicity: \(f > h\)
  \(\text{LDec} \in \text{deg 2}\)
  \(\text{hEval} > f\)

Use Partially hiding FE (PHFE) \textit{or} Special HE \textit{Shweta’s Talk!} 

[\text{AJS18}] \quad [\text{AR17,Agr18,LM18,JLS19}]
PHFE: FE for Simplicity

\[\text{deg } O(1), \text{ Public}\]

\[h(hct, s, sd):\]
1. \(hct_f \leftarrow \text{hEval}(f, hct)\)
2. \(y + 2e = \text{LDec}(s, hct)\)
3. \(y + 2e + 2\text{PFG}(sd)\)

\[\text{deg } 2, \text{ Private}\]

- Public input \(A = hct\)
- Private Input \(B = (s, sd)\)

Reveals output \(y + 2e + 2\text{PFG}(sd)\) and public input \(A = hct\)
Bilinear-based (PH)FE computes $g_T^{\text{output}}$

$\Rightarrow$ output $= y + 2e + 2\text{PFG}(sd)$ must be small to be extracted

Open: FE for large outputs

Weaken PFG

PFG: $e + \text{PFG}(sd)$ hides $e$

$\Rightarrow$ PFG$(sd)$ must be large
Small $\text{PFG}(sd)$ CANNOT smudge $e$ completely

\[ (r + e, e) \not\approx (r + e, e') \]

$\forall r_i \leftarrow \text{poly-bounded distribution}$

\[ r_i \not\approx r_i + 1 \]

E.g. \[ e_i \in \{0,1\} \]

**Bad, $e_i$ revealed**

\[ r_i + 0 \leftarrow U_{\{-B, \ldots, B\}} \quad B = \text{poly} \]

$-B$ \quad $B$

$-B + 1$ \quad $B + 1$
Best Possible: Small $\text{PFG}(sd)$ smudges $e$ partially

$I$, the set of bad coordinates

E.g. $r_i$ independent

Given $e + r$, $e_I$ hidden, $e_i$ revealed

Hope: $\neg \text{Bad}$, $e_i$ hidden

Bad, $e_i$ revealed

$r_i + 0 \leftarrow U_{\{-B, \ldots, B\}}$, $B = \text{poly}$
\( \Gamma \) is flawed-smudging: \( \forall \) small \( e \leftarrow E \) (poly \( B \)-bounded)

(Informal) \( \gamma + e \) hides \( e \) at all-but-a-few coordinates

in good case with \( 1/q(\lambda) \) probability

(formal) \( \exists I \) correlated with \( e, \gamma \)

\( (e, \gamma + e, I) \cong (e', \gamma + e, I) \quad e' \leftarrow E \mid e_I = e'_I \)

in good case with \( 1/q(\lambda) \) probability
PFG

\[ \text{PFG}(sd) + e \approx \gamma + e \]

- **PFG**
  - \( sd \)
  - \( r_i \), \( r \), \( r_j \), \( r_k \)

**Good with** \( 1/q(\lambda) \) **probability**
- \( e_I \) revealed & \( e_I \) hidden

**Strong PFG, \( I = \emptyset \)**
- nothing revealed & \( e \) hidden

**Bad with** \( 1 - 1/q(\lambda) \) **probability**
- all bets off, \( e \) revealed

\( r \)
At decryption, only $y + 2e + 2\text{PFG}(sd)$ revealed

Strong PFG

In good case w.p. $1/q(\lambda)$, FE secure, o.w., all bets off
Security Amplification via Bit-Fixing Secret Sharing [LM18]

Share($x$) → $x_1$ ... [Black square] ... $x_Q$

Eval Boolean $g^j$

Independently on shares

Eval($g^j, x_1$) Eval($g^j, x_i$) Eval($g^j, x_Q$)

$y_j^j = \text{Recon}(y_1^j, \ldots, y_i^j, \ldots, y_Q^j)$

Efficiency: $|x_i| \sim |g^j|$, independent of # of computations

Security: If one input share hidden, only $\{y_j^j\}$ revealed
Secret Sharing VBB

Share\((P) \rightarrow P_1 \rightarrow \ldots \rightarrow P_Q\)

Eval\((U(v^j,\ast), P_i)\)

Diff from HSS [BGI14, BGI15...]
Public output reconstruction

\[ y_1^j \rightarrow \ldots \rightarrow y_i^j \rightarrow \ldots \rightarrow y_Q^j \]

"Eval \(v^j\)"

Evaluator

\[ y^j = P(v^j) \]
Eval\((U(n^j,\cdot), P_i)\)

SS-VBB \sim \text{2-mesg MPC w/ reusable 1st mesg}

\[ \text{\textleftarrow Multikey FHE \textleftarrow LWE} \]

\[ \text{\textleftarrow Bilinear map [BL19]} \]
Security Amplification

\[ \frac{1}{q} \text{FE for P} \quad \text{bootstrap} \quad \frac{1}{q} \text{FE for NC}^0 \]

\[ Q \gg q, \text{w.h.p. some instance } i \text{ is secure} \]

\[ sk'(f): \quad sk_1(g) \quad sk_i(g) \quad sk_Q(g) \]
\[ g = \text{Eval}(f,*) \]

\[ ct'(x): \quad ct_1(x_1) \quad ct_i(*) \quad ct_Q(x_Q) \]

By SS-VBB, \( x \) hidden

\[ y = \text{Recon}(\{y_1\}, \{y_i\}, \{y_Q\}) \]
IO

+ Block-local PRG

FE

1-key, compact, NC0

(Strong) PFG

deg(O(1),2) PHFE

SS-VBB

Candidates
Comparison w/ ΔRG

LWE

bilinear
Want: Candidate deg 2 poly over \( \mathbb{Z}_p \) w/ small flawed-smudging outputs

Random deg 2 multivariate poly

\[
\mathbf{r}_l = g_l(x, y) = \sum c_{i_1 i_2} x_{i_1} y_{i_2}
\]

degenerate to over \( \mathbb{Z} \)

with small coefficients and inputs e.g., small Gaussian

Random deg 3 multivariate poly

\[
\mathbf{r}_l = g_l(x, y, z) = \sum c_{i_1 i_2 i_3} x_{i_1} y_{i_2} z_{i_3}
\]

over \( \mathbb{Z} \)
The AJS18 Idea

Random deg-3 multivariate polynomial

\[ r_l = g_l(x, y, z) = \sum c_{i_1i_2i_3} x_{i_1} y_{i_2} z_{i_3} \quad \text{over } Z \]

with small coefficients and inputs, e.g., small Gaussian

2-linear Map?

Hide \( y, z \) using 2-linear map, hide \( x \) as LWE noise

\[ C = A, As' + x, \quad Cs = x \quad \text{for } s = (-s', 1) \]

\[ g(x, y, z) = g(Cs, y, z) = h(C, y \otimes s, z) \]

Public seed  Private seed

Computable by deg (1,2)-PHFE
Hardness Assumptions [AJS18, JLMS19]

Given $C = A, As' + x$, $g(x, y, z) + e$ hides $e$ partially

1. $g(x, y, z) + e$ hides $e$ partially

2. LWE Leakage Assumption

$C = A, As' + x$, $g(x, y, z)$

$\approx$

$C = A, As' + x'$, $g(x, y, z)$

Naturally Generalize to Constant Degree $g$
Strong PFG v.s. $\Delta$RG

Strong PFG

$r \approx \gamma \leftarrow \Gamma$

$(e, e + \gamma) \equiv (e', e + \gamma)$

where $e' \leftarrow E$

with probability $1$/poly

$\Delta$RG

$(e, e + r) \approx_{Adv \text{t}} (e', e + r)$

where $e' \leftarrow E$

advantage $Adv \text{t} < 1-1$/poly

Degree 2 over $\mathbb{Z}_p$

$r$ small

$r$ smudges LWE noises partially
PFG v.s. $\Delta$RG

Degree 2 over $\mathbb{Z}_p$

$r$ small

$r$ smudges LWE noises partially

---

**PFG**

$r \approx \gamma \leftarrow \Gamma$

$(e, e + \gamma, I) \equiv (e', e + \gamma, I)$

where $e' \leftarrow E | e'_I = e_I$

with probability $1/\text{poly}$

---

**$\Delta$RG**

$(e, e + r) \approx_{\text{Adv}} (e', e + r)$

where $e' \leftarrow E$

advantage $Adv < 1 - 1/\text{poly}$

---

Different Security Amplification
1-key, compact, NC0

PHFE for
- deg (1,2) [AJS18,LM18]
- deg (O(1),2) [JLMS19]
- (NC¹, deg 2) [JLS19]

SS-VBB
- from MKFHE [LM18]
- from bilinear map [BL19]

SHE for
- deg O(1) [BV12]
- NC¹ from RLWE [AR17]
- for P [JLS19] from LWE inspired by [GVW15]
Thank you!

Questions?

Answers may be obfuscated