Black-box and Non-black-box Lower Bounds on Assumptions behind IO

Sanjam Garg (Berkeley)
Mohammad Mahmood (Univ. of Virginia)
Ameer Mohammed (Univ. of Virginia)
Indistinguishability Obfuscation (IO) [BGIRSVY01, GGHRSW13]
What primitive do you want?

- Functional Encryption: [Garg-Gentry-Halevi-Raykova-Sahai-Waters 2013]
- Witness Encryption: [Garg-Gentry-Sahai-Waters 2013]
- 2-round MPC: [Garg-Gentry-Halevi-Raykova 2013]
- Re-using garbled circuits: [Gentry-Halevi-Raykova-Wichs 2014]
- Deniable Encryption, KEM, Oblivious Transfer,...: [Sahai-Waters 2014]
- Random oracle instantiation: [Hohenberger-Sahai-Waters 2014]
- Secret sharing: [Komargodski-Naor 2014]
- 2-round adaptively-secure MPC: [Garg-Polychroniadou 2015]
What assumptions give us IO?
Figure 4: The map of different ways towards achieving iO for circuits in P/poly at the date of writing.

Picture by [Horváth, L Buttyán 16]
Can we use “standard assumptions”?
Main Results - Informal

**Thm:** Assuming OWFs and that Poly-Hierarchy does not collapse, none of primitives below imply IO in a `non-black-box` way:

- Witness encryption
- Predicate encryption
- Fully hom encryption

`Short output’ functional encryption` [GMM 17]

Previous Results: [MMNPS16]
Full black-box separation from OWF, CRH, IBE
• **Question:** Why is the result conditional?

• **Answer:** If $P = NP \rightarrow$ **statistically secure IO** for $P/poly$
  $\rightarrow$ Black-box IO possible by ignoring primitive $P$
Plan

1. Black-box model and its “non-bb extension”

2. Recipe for lower bounds for IO.

3. Separating IO from “short output” FE
Plan

1. Black-box model and its “non-bb extension”

2. Recipe for lower bounds for IO.

3. Separating IO from “short output” FE
Black-Box Framework [IR’89, RTV’04]

Natural when $P$ : OWF or TDP
How about self-feeding $P$?

Not black-box according to [IR, RTV]
But we do this sometimes..
Examples of where this trick is used

• FHE bootstrapping [Gentry’09]

• FE $\rightarrow$ IO [AJ’16,BV’16]
Let’s give it a name: **extended** black-box

- Inspired by [BKSY11, AS15, AS16] who allowed OWF gates
- Extended black-box: **all subroutines of primitive** are allowed
Relation to fully BB

- Extended black-box construction from P
- Fully black-box use of extended version of P
Main Results – Half Formal

*Thm:* Assuming OWFs and that Poly-Hierarchy does not collapse, none of primitives below imply IO in extended black-box way:

- Witness encryption
- Predicate encryption
- Fully hom encryption

\[\text{[GMM Crypto 17]}\]

- `Short output’ functional encryption [GMM 17]

\[\text{[GMM 17]}\]
Plan

1. Black-box model and its “extensions”

2. Recipe for lower bounds for IO.

3. Separating IO from “short output” FE
General technique: oracle separation

Separating Oracle

\[ P \rightarrow O \rightarrow IO^0 \]

\[ P^0 \text{ is secure} \]

Break
Recipe of attacking $\text{IO}^P$ in idealized model $\mathcal{P}$

1. [CKP’15] Compile out $\mathcal{P}$ from $\text{IO}^P \rightarrow$ get approx $\text{IO}$

2. [BBF’16] there is always an unbounded attack to approx $\text{IO}$

3. Combine two steps above $\rightarrow$ poly-query attack to $\text{IO}^P$

Only correct on 99% of inputs
Closer look at compiling out an oracle $\mathcal{P}$

We are here:

$\text{IO}^\mathcal{P}$

IO in $\mathcal{P}$ Model

Our Goal is:

$\text{IO}'$

“approximate IO” in plain model

How to obfuscate?
How to evaluate?
First try: emulate $\mathcal{P}$ on demand

- It is “secure” but $\mathcal{P}(x)$ and $\mathcal{P}(x)$ might be inconsistent.
- If we reveal $\mathcal{P}(x)$ to $B'$ for correctness $\rightarrow$ breaks security.
[CKP’15]: revealing useful `simulatable' queries

How to obfuscate? IO′(C)
What is the challenge?

- **Security**: can be simulated in ideal world of $IO^P$ so revealing it does not hurt the security of $IO$

- **Challenge**: to prove approximate correctness of $B'$ in plain model
If we compile out random oracle $\mathcal{P} \rightarrow$ get separation from OWF, CRH, etc.

- covers queries of $IO^P$ likely to be asked by $B'(x)$ (with error < 0.01)

- Any other query could be answered at random!
Plan

1. Black-box model and its “extensions”

2. Recipe for lower bounds for IO. Case of OWFs

3. Separating IO from “short output” FE
Functional Encryption

- Setup($1^κ$) $\rightarrow$ (PK, SK)
- Enc(PK, $m$) $\rightarrow$ ct
- KeyGen(SK, $f$) $\rightarrow$ Key$_f$  
  $f$ is arbitrary circuit
- Dec(ct, Key$_f$) = $f(x)$
- Security: $f(m_0) = f(m_1)$ $\rightarrow$ (PK, Key$_f$, ct$_0$) $\approx$ ind (PK, Key$_f$, ct$_0$)
**Thm:** Assuming OWFs and that Poly-Hierarchy does not collapse, none of primitives below imply IO in `extended` black-box way:

- **Short output** functional encryption [GMM 17]

- **Short output:** $|f(x)| < |ct| - \omega(|m|)$
- LWE-based FE of [GKPVZ13] satisfies this condition
- Positive results of [BV,AJ’15] use long outputs $|f(x)| \approx 2 \cdot |ct|$
Extended Functional Encryption

\[ \text{FE} = (\text{Setup, KeyGen, Enc, Dec}) \]

- **Extended** Black-Box use of Functional Encryption:
  Construction can use \( f^{\text{FE}} \) with all possible \text{FE} gates

- Equivalent to **fully black-box** use of \text{Extended FE} where we allow issuing keys for \( f^{\text{FE}} \) with all possible \text{FE} gates
Recall the goal: compiling out an ideal ext-FE oracle from any IO construction

We are here:

\[ \text{IO}^\text{FE} \]

IO: idealize \( FE \) Model for extended Func Enc

Our Goal is:

\[ \text{IO}' \]

“approximate IO” in \textit{plain} model
Enough to just compile out $\text{Dec}(\cdot)$ queries:

- $\text{Setup}(1^\kappa) \rightarrow (PK, SK)$
- $\text{Enc}(PK, m) \rightarrow ct$
- $\text{KeyGen}(SK, f) \rightarrow \text{Key}_f$
- $\text{Dec}(ct, \text{Key}_f) = f(x)$

Just a random oracle!
\( \mathcal{P} \): ideal ex-FE

Goal: compiling out Dec queries

**Challenge:**

- Any Dec\((ct, f)\) query has its own internal queries during \( f^{FE}(m) \)
- Queries are not simulatable \(\rightarrow\) not OK to be passed to \( B' \)
Idea 1: if we know $m$ inside $ct = \text{Enc}(m)$ $\rightarrow \text{Dec}(ct, f)$ turns into $\text{Dec}(ct, f)$ because we can run $f^{\text{FE}}(m)$ instead.

Idea 2: we can assume every $ct$ is decrypted at most once.

Final goal: show that $\text{Dec}(ct, f)$ does not happen during final exec $B'$.

$\mathcal{P}$: ideal ex-FE

Goal: compiling out Dec queries.
Final Idea (using short output of FE):
learner sees a fixed polynomial number of $\text{Dec}(\text{ct}, f)$ queries

- By choosing $t$ large enough $\Rightarrow$ no “unknown” ciphertext during final exec

$\mathcal{P}$: ideal ex-FE
Goal: compiling out Dec queries
Short output $\rightarrow$ only poly new unknown ciphertexts

- Suppose $|f(x)| \ll |ct| - |m|$
  where $ct = \text{Enc}(m)$ and $f(x) = \text{Dec}(ct)$

- **Claim**: If we use random $\text{enc} : \{0,1\}^{|m|} \rightarrow \{0,1\}^{|ct|}$, then any algorithm $A$ with $s$ bits of `advice’ can hit only at most $s$ “unknown” ciphertexts

- **Proof**:
  1. A string $ct$ is a valid ciphertext with probability $2^{|m| - |ct|}$
  2. “Hitting” a valid ciphertext needs $\approx |ct| - |m|$ bits of `advice’
  3. The answer $f(x)$ can only give back $|f(x)|$ bits of advice
  4. If $|f(x)| < |ct| - |m|$ $\rightarrow$ after $t$ steps we run out of advice bits!
Recap

*Thm:* Assuming OWFs and that Poly-Hierarchy does not collapse, none of primitives below imply IO in `extended black-box` way:

- Witness encryption
- Predicate encryption
- Fully hom encryption
- Short output functional encryption

[GMM Crypto 17]

[GMM 17]
Future Directions?

• Tighter upper and lower bounds for output length of FE for IO?

• Long output FE from LWE?

• Revisiting classical separation results like OWF $\Rightarrow$ PKE [IR’89] even more important in light of recent IBE from DDH [DG’17]
Thanks!