How to Obfuscate MPC Inputs

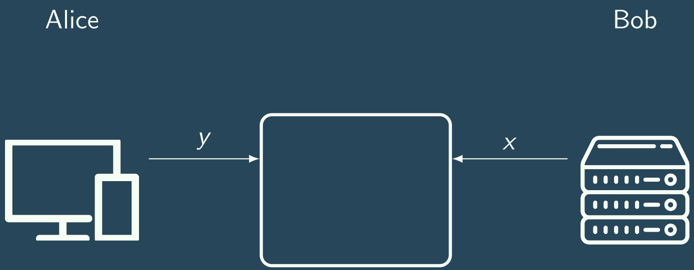
Theory of Cryptography Conference 2022

School of Electrical Engineering and Computer Science
Oregon State University

Ian McQuoid Mike Rosulek Jiayu Xu

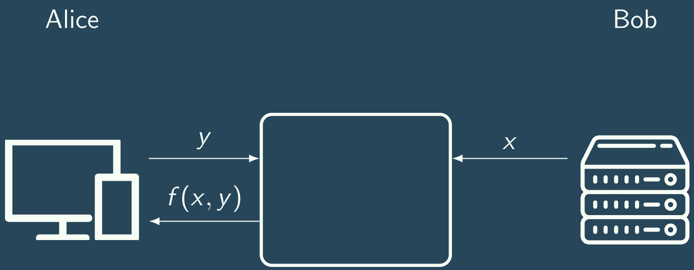
Introduction

- Bob wants to provide a service to Alice using his input x.



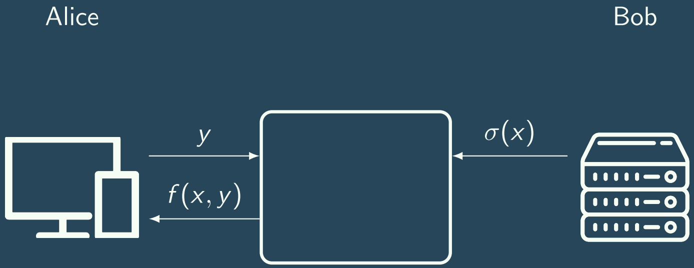
Introduction

- Bob wants to provide a service to Alice using his input x.
- But both Alice's and Bob's inputs contain private data.



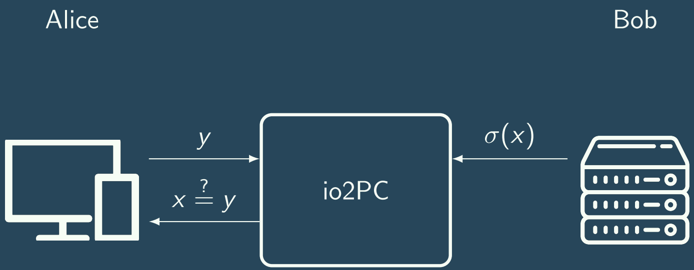
Introduction

- Bob wants to provide a service to Alice using his input x.
- But both Alice's and Bob's inputs contain private data.
- Bob is worried about compromise of his service leaking x.



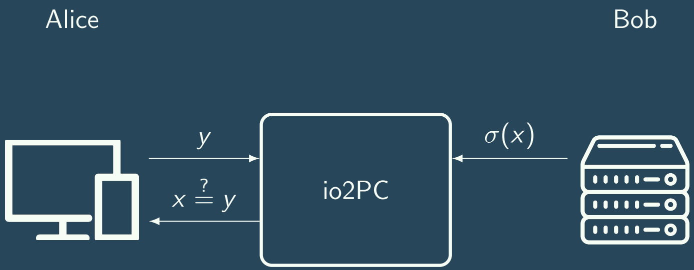
io2PC- Point Functions

- Evaluating $x \stackrel{?}{=} y$ online is oblivious and interactive.



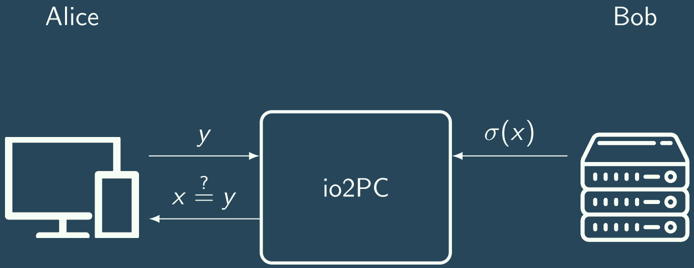
io2PC- Point Functions

- Evaluating $x \stackrel{?}{=} y$ online is oblivious and interactive.
- On compromise, we only leak an equality oracle $y \mapsto x \stackrel{?}{=} y$.



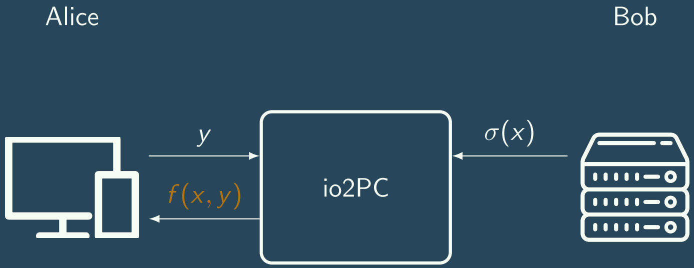
io2PC- Point Functions

- Evaluating $x \stackrel{?}{=} y$ online is oblivious and interactive.
- On compromise, we only leak an equality oracle $y \mapsto x \stackrel{?}{=} y$.
- Offline evaluation of $x \stackrel{?}{=} y$ must be done *post-compromise*.



io2PC- General Case

- Evaluating $f(x, y)$ online is oblivious and interactive.
- Only an oracle $f(x, \cdot)$ is leaked on Bob's compromise.
- Offline evaluation of $f(x, y)$ must be done *post-compromise*.



io2PC

Theorem

There exists an UC-secure io2PC protocol for a function f, if the related class of functions $\mathcal{C}_f = \{f(x, \cdot) \mid x \in \{0,1\}^n\}$ has a VBB obfuscation in either the random oracle or generic group models.

io2PC

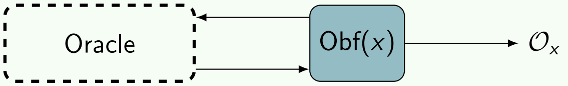
Theorem

There exists an UC-secure io2PC protocol for a function f, if the related class of functions $\mathcal{C}_f = \{f(x, \cdot) \mid x \in \{0,1\}^n\}$ has a VBB obfuscation in either the random oracle or generic group models.

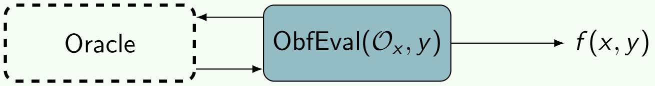
- We achieve this by replacing the corresponding non-interactive oracle queries with interactive protocols.

Virtual Black-Box Obfuscation

Obfuscation



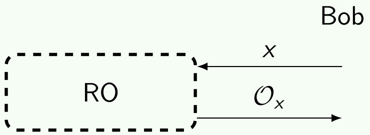
Evaluation



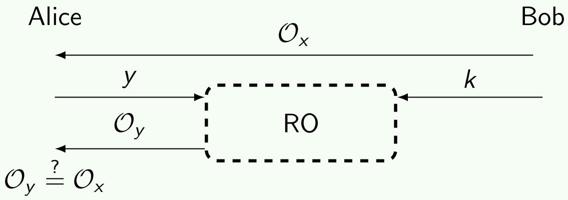
A VBB obfuscation $\mathcal{O}_x$ can be *simulated* with only oracle access to $f(x, \cdot)$.

Virtual Black-Box Obfuscation - Point Function

Obfuscation

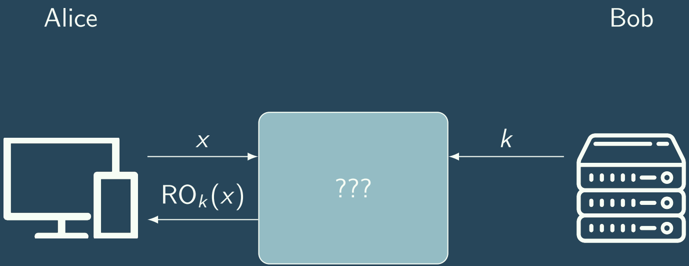


Evaluation



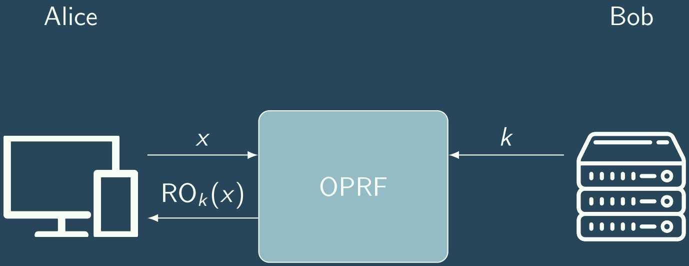
Interactive Random Oracles

What does an “interactive random oracle” look like?



Oblivious Psuedorandom Functions

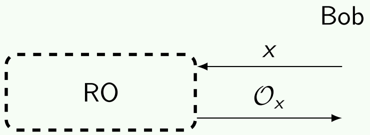
What does an “interactive random oracle” look like?



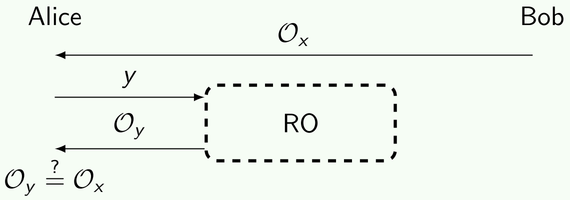
- JKKX16 provides an oblivious psuedorandom function (OPRF) achieving this property!

Input Obfuscation in the Random Oracle Model

Obfuscation

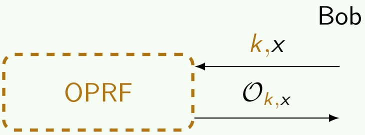


Evaluation

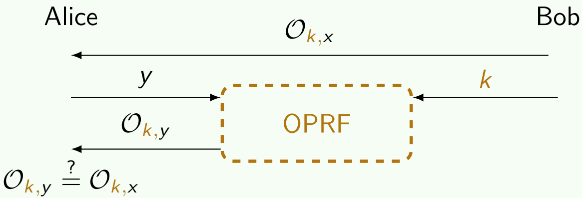


Input Obfuscation in the Random Oracle Model

Obfuscation

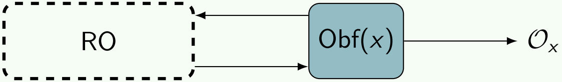


Evaluation

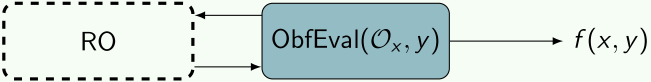


Input Obfuscation in the Random Oracle Model

Obfuscation

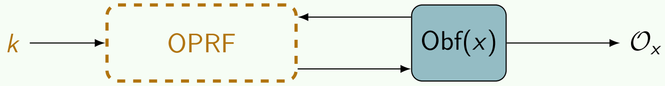


Evaluation

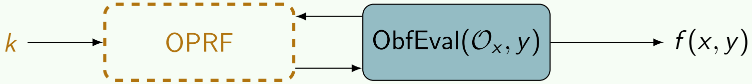


Input Obfuscation in the Random Oracle Model

Obfuscation



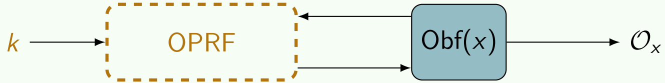
Evaluation



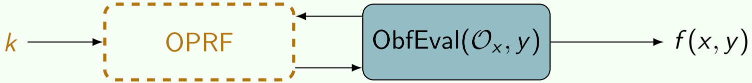
- Why is this not trivial in 2pc? — The idealized primitives are exponential in size!

Input Obfuscation in the Random Oracle Model

Obfuscation



Evaluation



- Why is this not trivial in 2pc? — The idealized primitives are exponential in size!

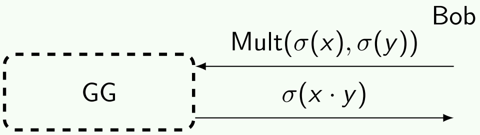
Can we construct interactive versions of other idealized primitives?

What about generic groups?

Generic Groups

For a uniform encoding $\sigma : \mathbb{Z}_p \to \{0,1\}^*$

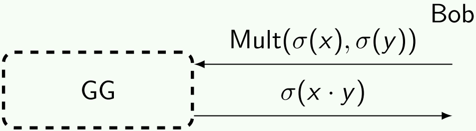
Addition



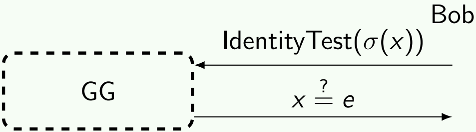
Generic Groups

For a uniform encoding $\sigma : \mathbb{Z}_p \to \{0,1\}^*$

Multiplication



Identity Test



Interactive Generic Groups

- Given a publicly accessible GG $\mathcal{G} := (g, \cdot)$ and a "key" $(k \leftarrow \mathcal{K}, \hat{g} \leftarrow \mathcal{G})$.
- We construct an iGG where operations are interactive, oblivious, and require the key.
- Elements take the form $(F_k(m), \hat{g}^x \cdot g^m)$.

Interactive Generic Groups

- Given a publicly accessible GG $\mathcal{G} := (g, \cdot)$ and a "key" $(k \leftarrow \mathcal{K}, \hat{g} \leftarrow \mathcal{G})$.
- We construct a iGG where operations are interactive, oblivious, and require the key.
- Elements take the form $(F_k(m), \hat{g}^x \cdot g^m)$.

Multiplication

$$(F_k(m_1), g_1) \cdot_k (F_k(m_2), g_2) := (F_k(m_3), \hat{g}^{x_1+x_2} \cdot g^{m_3})$$

Interactive Generic Groups

- Given a publicly accessible GG $\mathcal{G} := (g, \cdot)$ and a "key" $(k \leftarrow \mathcal{K}, \hat{g} \leftarrow \mathcal{G})$.
- We construct an iGG where operations are interactive, oblivious, and require the key.
- Elements take the form $(F_k(m), \hat{g}^x \cdot g^m)$.

Multiplication

$$(F_k(m_1), g_1) \cdot_k (F_k(m_2), g_2) := (F_k(m_3), \hat{g}^{x_1+x_2} \cdot g^{m_3})$$

- $\hat{g}^{x_1+x_2} \cdot g^{m_3}$ can be computed using the public group.

Interactive Generic Groups

- Given a publicly accessible GG $\mathcal{G} := (g, \cdot)$ and a "key" $(k \leftarrow \mathcal{K}, \hat{g} \leftarrow \mathcal{G})$.
- We construct an iGG where operations are interactive, oblivious, and require the key.
- Elements take the form $(F_k(m), \hat{g}^x \cdot g^m)$.

Addition

$$(F_k(m_1), g_1) \cdot_k (F_k(m_2), g_2) := (F_k(m_3), \hat{g}^{x_1+x_2} \cdot g^{m_3})$$

- $\hat{g}^{x_1+x_2} \cdot g^{m_3}$ can be computed using the public group.
- $F_k(m_3)$ can be computed *interactively* in 2PC.

Personalized Generic Groups

- Given a publicly accessible GG $\mathcal{G} := (g, \cdot)$ and a "key" $(k \leftarrow \mathcal{K}, \hat{g} \leftarrow \mathcal{G})$.
- Elements take the form $(F_k(m), \hat{g}^x \cdot g^m)$.

Identity Test

$$\mathsf{IdentityTest}((F_k(m), g_1)) := g_1 \stackrel{?}{=} g^m$$

Interactive Generic Groups

- Given a publicly accessible GG $\mathcal{G} := (g, \cdot)$ and a "key" $(k \leftarrow \mathcal{K}, \hat{g} \leftarrow \mathcal{G})$.
- Elements take the form $(F_k(m), \hat{g}^x \cdot g^m)$.

Identity Test

$$\mathsf{IdentityTest}((F_k(m), g_1)) := g_1 \stackrel{?}{=} g^m$$

- $g_1 \stackrel{?}{=} g^m$ can be calculated using the public group.

Interactive Generic Groups

- Given a publicly accessible GG $\mathcal{G} := (g, \cdot)$ and a "key" $(k \leftarrow \mathcal{K}, \hat{g} \leftarrow \mathcal{G})$.
- Elements take the form $(F_k(m), \hat{g}^x \cdot g^m)$.

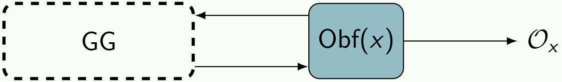
Identity Test

$$\mathsf{IdentityTest}((F_k(m), g_1)) := g_1 \stackrel{?}{=} g^m$$

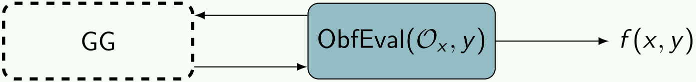
- $g_1 \stackrel{?}{=} g^m$ can be calculated using the public group.
- Alice *interactively* learns blindings ${g_1}^b$ and g^{bm} which she compares.

Input Obfuscation in the Generic Group Model

Obfuscation

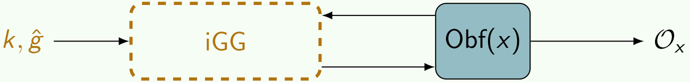


Evaluation

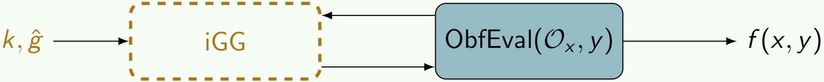


Input Obfuscation in the Generic Group Model

Obfuscation



Evaluation



Conclusion

- We introduce the study of input obfuscation for secure two-party computation (io2PC).
- We provide a compiler from VBB in the GGM and ROM to io2PC.
- To construct the latter, we provide an oblivious, interactive GG analogous to an OPRF.
- We provide explicit io2PC protocols for point functions and hyperplane membership using our compiler.

Conclusion

- We introduce the study of input obfuscation for secure multi-party computation (io2PC).
- We provide a compiler from VBB in the GGM and ROM to UC-secure io2PC.
- To construct the latter, we provide an oblivious, interactive GG analogous to an OPRF.
- We prove that known VBB obfuscations of point functions and hyperplane membership are compatible.
- We conjecture that io2PC is possible for generic graded encodings and therefore all $\mathcal{P}$.