Reach Restricted Reactive Program Obfuscation

And its Application to MA-ABE

Kaartik **B**hushan, Sai Lakshmi Bhavana **O**bbattu, Manoj **P**rabhakaran, **Rajeev Raghunath**

April 14, 2024

Some Surprising Results

Laconic Oblivious Transfer from DDH [CDG+17]

Identity Based Encryption (IBE) from DDH [DG17a, DG17b]

2-round MPC from 2-round OT [BL17, GS17]

Some Surprising Results

Laconic Oblivious Transfer from DDH [CDG+17]

Identity Based Encryption (IBE) from DDH [DG17a, DG17b]

2-round MPC from 2-round OT [BL17, GS17]

Key Ingredient in all the above: Garbled Circuit Chaining

Some Surprising Results

Laconic Oblivious Transfer from DDH [CDG+17]

Identity Based Encryption (IBE) from DDH [DG17a, DG17b]

2-round MPC from 2-round OT [BL17, GS17]

Key Ingredient in all the above: Garbled Circuit Chaining

But no common abstraction or framework!

Some Surprising Results

Laconic Oblivious Transfer from DDH [CDG+17]

Identity Based Encryption (IBE) from DDH [DG17a, DG17b]

2-round MPC from 2-round OT [BL17, GS17]

Key Ingredient in all the above: Garbled Circuit Chaining

But no common abstraction or framework!

Motivation Behind This Work: Abstract the powerful ingredient driving these results (and beyond)

Some Surprising Results

Laconic Oblivious Transfer from DDH [CDG+17]

Identity Based Encryption (IBE) from DDH [DG17a, DG17b]

2-round MPC from 2-round OT [BL17, GS17]

Key Ingredient in all the above: Garbled Circuit Chaining

But no common abstraction or framework!

Motivation Behind This Work: Abstract the powerful ingredient driving these results (and beyond) *as Obfuscation!*

What is Obfuscation?

Program Obfuscation:

- keeping secrets in a program

What is Obfuscation?

Program Obfuscation:

- keeping secrets in a program
- even against an adversary that captures the entire computer on which it is run
- without any trusted hardware

Subtle to formalize

Obfuscation Results

Different notions of Obfuscation

- Virtual Black-Box Obfuscation [BGI+01]
- Indistinguishability Obfuscation (iO) [BGI+01, JLS20]
- Average Case Obfuscation [HRsV07]
- Virtual Grey Box Obfuscation [BCTKP14]
- Differing Inputs Obfuscation [ABG+13] and public-coin DiO [IPS14]

Require strong assumptions in general (if not impossible)

Obfuscation Results

Different notions of Obfuscation

Virtual Black-Box Obfuscation [BGI+01]
Indistinguishability Obfuscation (iO) [BGI+01, JLS20]
Average Case Obfuscation [HRsV07]
Virtual Grey Box Obfuscation [BCTKP14]
Differing Inputs Obfuscation [ABG+13] and public-coin DiO [IPS14]

Require strong assumptions in general (if not impossible)

- Obfuscation achievable from standard assumptions, when programs are sampled in a customized fashion:
  - Obfuscation for Re-Encryption [HRsV07] - from DDH
  - Obfuscation for Evasive Functions [BBC+13] - from DDH variant
  - Obfuscation for Compute-and-Compare functions [WZ17] - from LWE
  - **In this work:** For programs sampled interactively, enforcing a restriction on what information the adversary has about its contents.

A Motivating Example

A program that contains a message and an encryption public-key PK.

If a valid decryption key SK is given as input, it outputs the message.

A Motivating Example

A program that contains a message and an encryption public-key PK. If a valid decryption key SK is given as input, it outputs the message.

Naïve idea: *Encrypting the message using PK will be an obfuscation of this program!*

A Motivating Example

A program that contains a message and an encryption public-key PK. If a valid decryption key SK is given as input, it outputs the message.

Naïve idea: *Encrypting the message using PK will be an obfuscation of this program!*

Can be turned into a valid notion of obfuscation:

Interactive sampling of the program: (PK,SK) pairs are generated secretly. Each SK can be published fully, or not revealed at all, as requested by the adversary.

Simulation of Obfuscated Program: If SK published, an adversary is allowed to learn the message — from which a valid ciphertext can be constructed. If SK not published, a random ciphertext is a valid simulation of the real ciphertext.

Conversely, such an obfuscation yields PKE.

Our Goal

A definition that formalizes similar seemingly naïve ideas of obfuscation

Example: IBE as Obfuscation
Ciphertext is the obfuscation of the following program:

Hardwired: message m, identity id, a signature verification key VK

On input σ: if σ is a valid signature on id w.r.t. VK, output m.

Issue a decryption key for id by simply signing id

Reactive Program

Let Σ, M be the space of states and messages respectively.

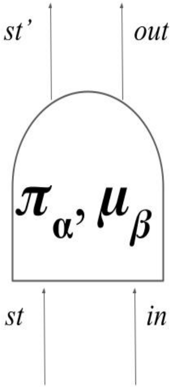
Reactive Program $P = (\pi_\alpha, \mu_\beta)$**:**

Transition function $\pi_\alpha : \Sigma \times \mathcal{X} \to \Sigma$
Message function $\mu_\beta : \Sigma \to M$
(for some hardwired secrets α, β)

Evaluating a Reactive Program:

$P(\mathsf{st}, in)$:

$$\mathsf{st}' = \pi_\alpha(\mathsf{st}, in)$$
$$out = \mu_\beta(\mathsf{st}')$$



Reach-Restricted Reactive Program

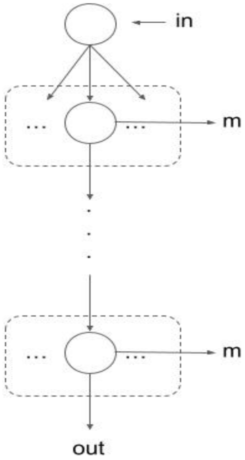
We require a partition of the state space,
$\Sigma = \Sigma_1 \cup \cdots \cup \Sigma_n$

A typical example will have:

- $O(\kappa)$ parts
- $2^{O(\kappa)}$ states in each part.

where κ is the security parameter.

The parts should form a tree.



Reach-Restricted Reactive Program

We require a partition of the state space, $\Sigma = \Sigma_1 \cup \cdots \cup \Sigma_n$

A typical example will have:

- $O(\kappa)$ parts
- $2^{O(\kappa)}$ states in each part.

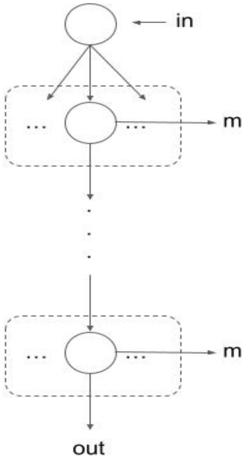
where κ is the security parameter.

The parts should form a tree.

Reach-Restriction:

Adversary can find inputs that take the program to at most one state in each part of the partition

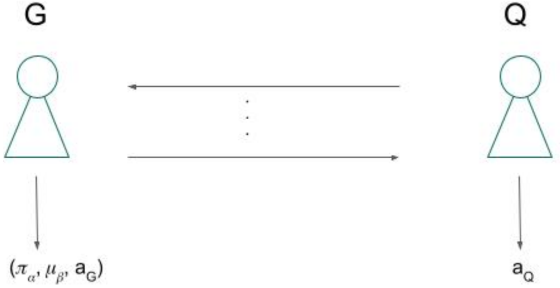
When the program is "sampled properly"



Interactive Sampling

Rules for sampling a program formalized as a class of
Reactive Program Generators

- A generator G interacts with an adversary Q
- Outputs a reactive program (π_α, μ_β).
  Also auxiliary information a_G, a_Q produced



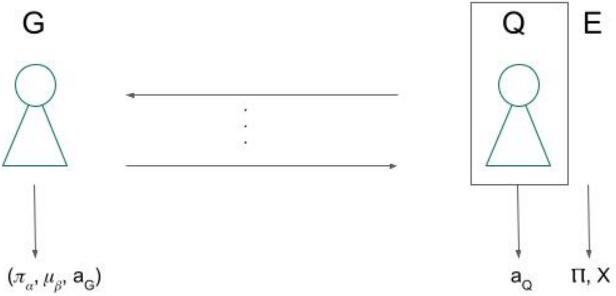
Formalizing Reach Restriction

To which all states can an adversary Q take a reactive program generated by a generator G (even given $(\pi_\alpha, \mu_\beta, \mathrm{a}_G;\ \mathrm{a}_Q)$)?

Formalizing Reach Restriction

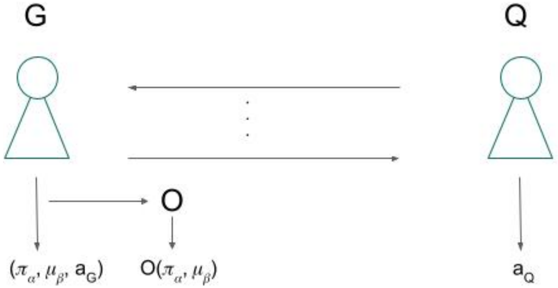
To which all states can an adversary Q take a reactive program generated by a generator G (even given $(\pi_\alpha, \mu_\beta, \mathrm{a}_G;\ \mathrm{a}_Q)$)?

- An extractor E can output all such states.
  - Encoded as an (idealized) reactive program Π and input sequences X for it, s.t. reachable states in π_α are reached in Π using X.
- Can have at most one state in each part in the state-space partition.



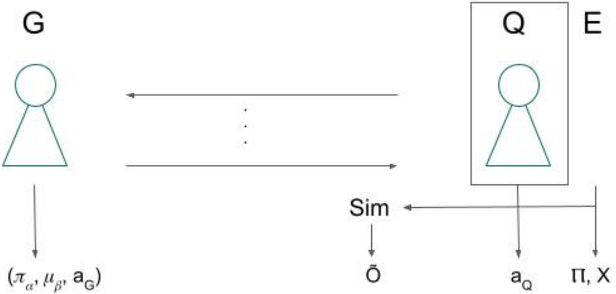
The Real World:

- G interacts with Q
- output of interaction: $(\pi_\alpha, \mu_\beta, \mathrm{a}_G;\ \mathrm{a}_Q)$
- Obfuscator $\mathcal{O}$ outputs $\mathcal{O}(\pi_\alpha, \mu_\beta)$



The Ideal World:

- G interacts with Q
- output of interaction: $(\pi_\alpha, \mu_\beta, \mathrm{a}_G;\ \mathrm{a}_Q)$, E outputs Π, X
- $\mathrm{Sim}\Big(\Pi, X, \{\mu_\beta(\mathrm{st}) \mid \mathrm{st} \in \Pi(X)\}\Big)$ outputs $\tilde{\mathcal{O}}$



Defining R3PO Security

A Strong Simulation-Based Definition

$\mathcal{O}$ is an R3PO scheme for $\mathcal{G}$ w.r.t. a class of adversaries $\mathcal{Q}$ if, $\forall G \in \mathcal{G}$ and $Q \in \mathcal{Q}$, there exists a simulator Sim s.t. Real World is indistinguishable from Ideal World

$$\left\{\mathcal{O}(\pi_\alpha, \mu_\beta),\ \mathrm{a}_G,\ \mathrm{a}_Q\right\} \approx \left\{\mathsf{Sim}\Big(\Pi, X, \{\mu_\beta(\mathsf{st}) \mid \mathsf{st} \in \Pi(X)\}\Big),\ \mathrm{a}_G,\ \mathrm{a}_Q\right\}$$

Example 1: Commitment Opening R3PO

Commitment Scheme

- $\mathsf{gen}(1^\kappa) \to \mathsf{crs}$
- $\mathsf{commit}(\mathsf{crs}, m) \to (c, d)$
- $\mathsf{open}(\mathsf{crs}, c, d) \to m$

Properties Required:

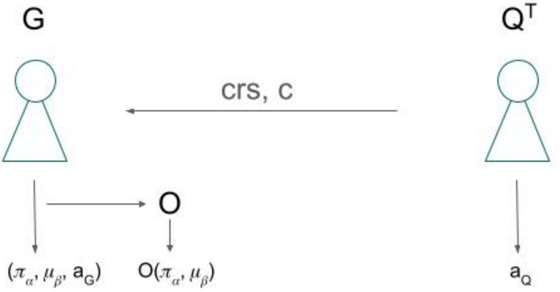
1 Computational Hiding: commitment c does not reveal message m.

2 Computational Binding: commitment c can be opened to at most a single message m. Further, there exists an extractor $\mathcal{E}$ that can extract this m.

Example 1: Commitment Opening R3PO

Interaction:

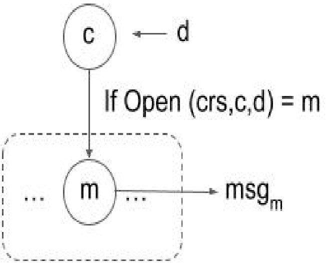
- Q^T gets *crs* from T and sends *crs*, c to G.



Example 1: Commitment Opening R3PO

Reactive Program:

$$\pi(\mathsf{st}_c^1, d) = \begin{cases} \mathsf{st}_m^2, \text{ if open}(\mathsf{crs}, c, d) = m \\ \perp, \text{ else} \end{cases}$$

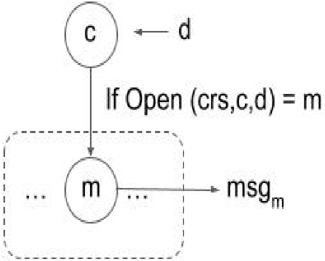


⋆ Interested in adversaries of the form Q^T that sample crs honestly.

Example 1: Commitment Opening R3PO

Reactive Program:

$$\pi(\text{st}_c^1, d) = \begin{cases} \text{st}_m^2, \text{ if open}(\text{crs}, c, d) = m \\ \perp, \text{ else} \end{cases}$$



⋆ Interested in adversaries of the form Q^T that sample crs honestly.

Theorem 1 (Informally)

If the DDH assumption holds, there exists a Commitment scheme and a R3PO for Commitment-Opening.

Example 2: Signature-Checking

Signature Scheme

- $\mathsf{gen}(1^\kappa) \to (\mathsf{vk}, \mathsf{sk})$
- $\mathsf{sign}(\mathsf{sk}, m) \to \tau$
- $\mathsf{verify}(\mathsf{vk}, m, \tau) \to \{0, 1\}$

Properties Required:

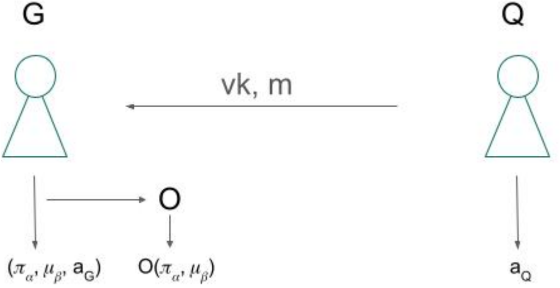
1 Correctness.

2 Unforgeability: without sk, hard to forge signature on a new message.

Example 2: Signature-Checking R3PO

Interaction:

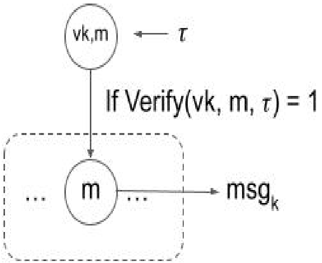
- Q sends vk, m to G.



Example 2: Signature-Checking R3PO

Reactive Program:

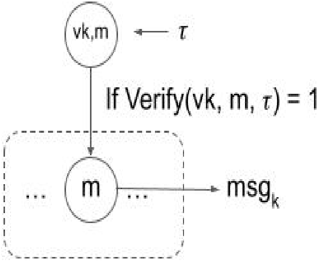
$$\pi(\mathsf{st}^1_{\mathsf{vk},m}, \tau) = \begin{cases} \mathsf{st}^2_m, \text{ if verify}(\mathsf{vk}, m, \tau) = 1 \\ \perp, \text{ else} \end{cases}$$



Example 2: Signature-Checking R3PO

Reactive Program:

$$\pi(\mathsf{st}^1_{\mathsf{vk},m}, \tau) = \begin{cases} \mathsf{st}^2_m, \text{ if } \mathsf{verify}(\mathsf{vk}, m, \tau) = 1 \\ \bot, \text{ else} \end{cases}$$



Theorem 2 (Informally)

If the DDH assumption holds, there exists a Signature scheme and a R3PO for Signature-Checking.

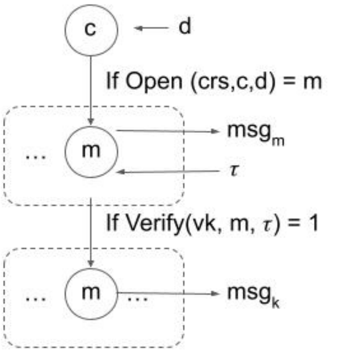
Towards R3PO of Larger Reactive Programs

Can we combine R3PO for Commitment Opening and Signature Checking?

Reactive Program:

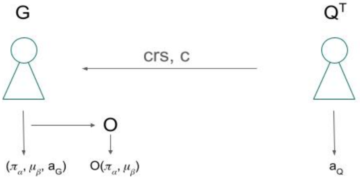
$$\pi(\mathsf{st}_c^1, d) = \begin{cases} \mathsf{st}_m^2, \text{ if } \mathsf{open}(\mathsf{crs}, c, d) = m \\ \perp, \text{ else} \end{cases}$$

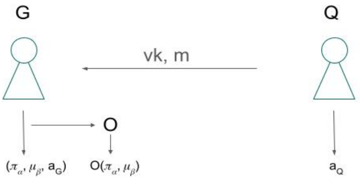
$$\pi(\mathsf{st}_m^2, \tau) = \begin{cases} \mathsf{st}_m^3, \text{ if } \mathsf{verify}(\mathsf{vk}, m, \tau) = 1 \\ \perp, \text{ else} \end{cases}$$

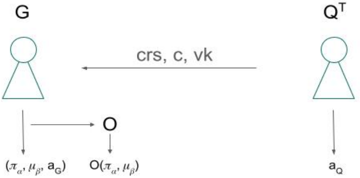


Towards R3PO of Larger Reactive Programs

Can we combine R3PO for Commitment Opening and Signature Checking?

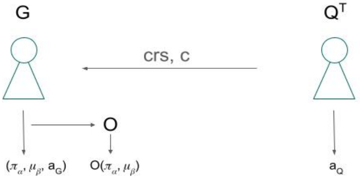


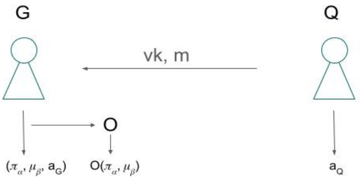


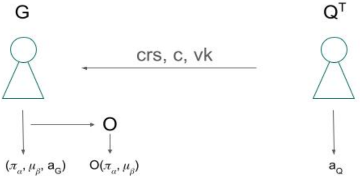


Towards R3PO of Larger Reactive Programs

Can we combine R3PO for Commitment Opening and Signature Checking?





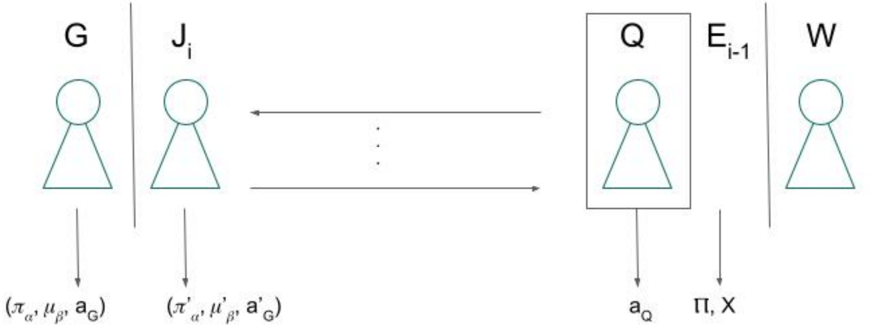


Need to be careful in handling the interaction!

Decomposition Property

We say that a generator class $\mathcal{G}$ **decomposes** to a generator class $\mathcal{G}_i$ (at partition i) if the following bi-simulations are indistinguishable.

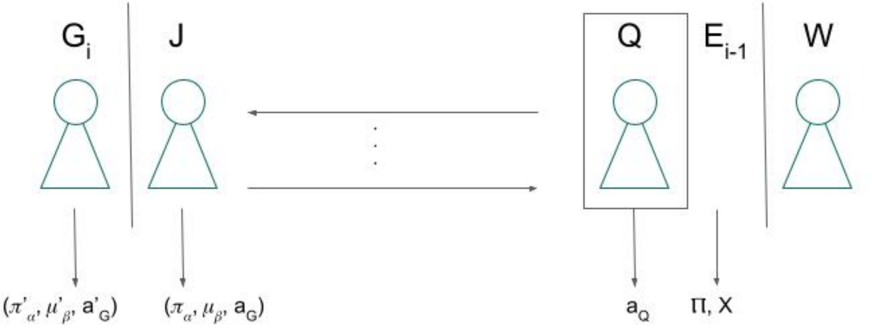
- In the interaction between $G \in \mathcal{G}$ and $Q \in \mathcal{Q}$, for all $(i-1)$-partial reach extractors E_{i-1}, there exists J_i, W s.t. J_i outputs $(\pi'_\alpha, \mu'_\beta, \mathrm{a}'_G)$.



Decomposition Property

We say that a generator class $\mathcal{G}$ **decomposes** to a generator class $\mathcal{G}_i$ (at partition i) if the following bi-simulations are indistinguishable.

- In the interaction between $G_i \in \mathcal{G}_i$ and $Q|E_{i-1}|W \in \mathcal{Q}$, there exists J that outputs $(\pi_\alpha, \mu_\beta, \mathrm{a}_G)$.



Composition Theorem

Theorem 3 (Informally)

Let $\mathcal{G}_1, \ldots, \mathcal{G}_n$ be generator classes with R3PO schemes $\mathcal{O}_1, \ldots, \mathcal{O}_n$. If a generator class $\mathcal{G}$ decomposes to $\mathcal{G}_i$ at each partition $i \in [n]$, then there exists a R3PO scheme for $\mathcal{G}$.

The proof uses garbled-circuit chaining.

Composition Theorem

Theorem 3 (Informally)

Let $\mathcal{G}_1, \ldots, \mathcal{G}_n$ be generator classes with R3PO schemes $\mathcal{O}_1, \ldots, \mathcal{O}_n$. If a generator class $\mathcal{G}$ decomposes to $\mathcal{G}_i$ at each partition $i \in [n]$, then there exists a R3PO scheme for $\mathcal{G}$.

Corollary:

If there exists a R3PO for commitment-opening, then:

- there exists a R3PO for sequence of commitment-openings
- there exists a 2-round MPC protocol secure against semi-honest dishonest majority corruption [BL17, GS17]

Composition Theorem

Theorem 3 (Informally)

Let $\mathcal{G}_1, \ldots, \mathcal{G}_n$ be generator classes with R3PO schemes $\mathcal{O}_1, \ldots, \mathcal{O}_n$. If a generator class $\mathcal{G}$ decomposes to $\mathcal{G}_i$ at each partition $i \in [n]$, then there exists a R3PO scheme for $\mathcal{G}$.

Corollary:

If there exists a R3PO for signature-checking, then:

- there exists a R3PO for sequence of signature-checkings
- there exists an adaptive-secure IBE scheme [DG17a, DG17b]

Composition Theorem

Theorem 3 (Informally)

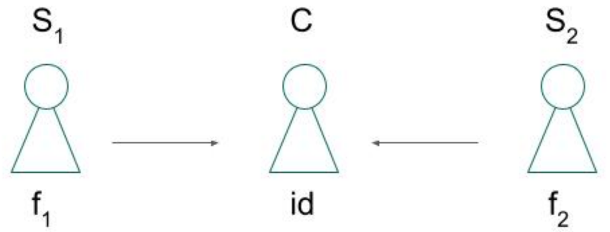
Let $\mathcal{G}_1, \ldots, \mathcal{G}_n$ be generator classes with R3PO schemes $\mathcal{O}_1, \ldots, \mathcal{O}_n$. If a generator class $\mathcal{G}$ decomposes to $\mathcal{G}_i$ at each partition $i \in [n]$, then there exists a R3PO scheme for $\mathcal{G}$.

Corollary:

If there exists a R3PO for commitment-opening and a R3PO for signature-checking, then:

- there exists a R3PO for commitment-opening followed by signature checking.
- if there exists an ABE scheme, there exists a "private" MA-ABE scheme (our work).

Prior Works for MA-ABE

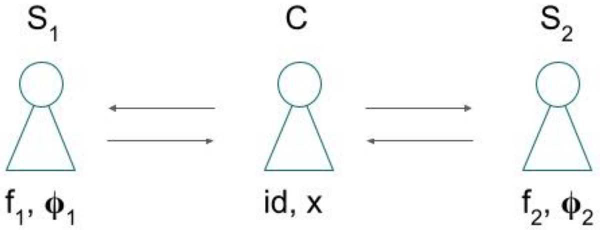


Global ID Model [Cha07, LW10]:

- Each client has a global id
- Only interaction: servers send credentials for id to a client
- Current results rely on the Random Oracle Model: E.g., [DKW20] for DNF formulae under the LWE assumption.

Private MA-ABE: A client can privately decide on the attributes it wants to acquire, as long as it conforms to the servers' policy. Client can send a message to each server first.

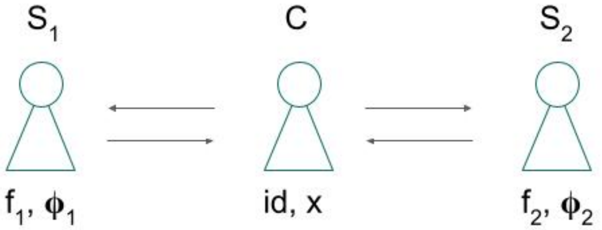
Attribute Verification



2-round protocol for Attribute Verification:

- Decentralized setup of Servers: publish global public keys.
- Round 1: C sends a request to S_1 and S_2.
- Round 2: Servers S_1 and S_2 send response to C.

Attribute Verification

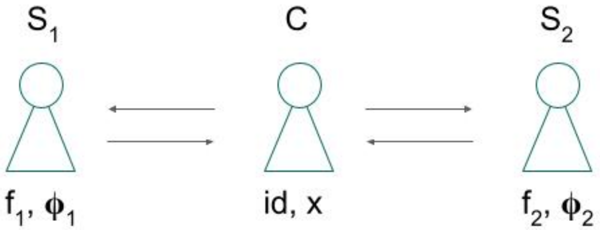


Completeness:

if $\Phi_1(\text{id}, x) = 1$ and $\Phi_2(\text{id}, x) = 1$, then C gets $f_1(\text{id}, x)$, $f_2(\text{id}, x)$.

Hiding: Server S_b learns nothing about (id, x) and Φ_{1-b}.

Attribute Verification



Solution using R3PO:

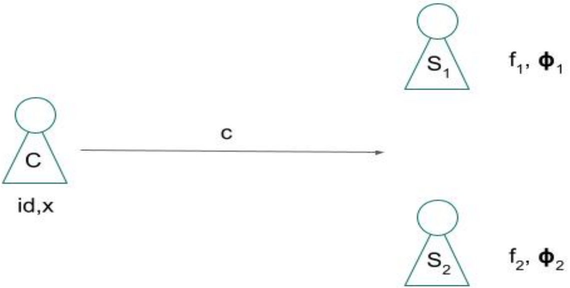
Use commitment to hide (id, x).

Use signatures to give proof of verification.

Attribute Verification

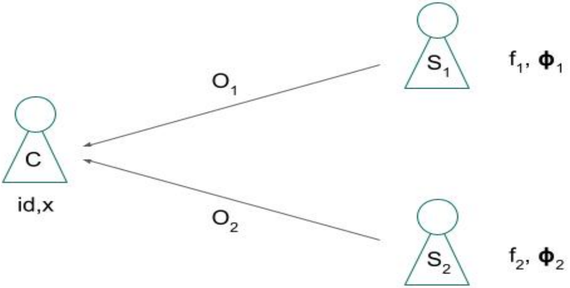
Round 1:

- Client C computes $(c, d) \leftarrow \text{commit}(\text{crs}, \text{id})$ and sends c to servers S_1, S_2.



Round 2**:**

- Server S_1 sends $\mathcal{O}_1$ to Client C.
- Server S_2 sends $\mathcal{O}_2$ to Client C.



Round 2:

- Server S_1 sends $\mathcal{O}_1$ to Client C.
- Server S_2 sends $\mathcal{O}_2$ to Client C.

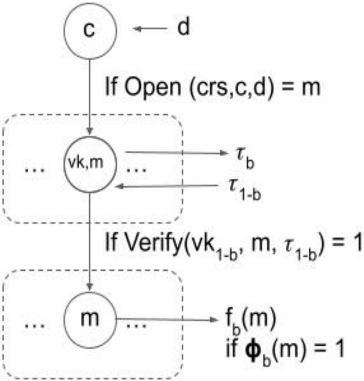
where, each $\mathcal{O}_i$ is R3PO of program with:

transition function π:

$$\pi(\mathsf{st}_c^1, d) = \mathsf{st}_m^2, \text{ if } \mathsf{open}(\mathsf{crs}, c, d) = m$$
$$\pi(\mathsf{st}_m^2, \tau) = \mathsf{st}_m^3, \text{ if } \mathsf{verify}(\mathsf{vk}_{1-b}, m, \tau_{1-b}) =$$

message function μ:

$$\mu_{\mathsf{sk}_b, f_b, \Phi_2}(\mathsf{st}_m^3) = \mathsf{sign}(\mathsf{sk}_b, m)$$
$$\mu_{\mathsf{sk}_b, f_b, \Phi_b}(\mathsf{st}_m^3) = f_b(m), \text{ if } \Phi_b(m) = 1$$



Attribute Verification

Theorem 4

If there exists an R3PO for commitment-opening and signature-checking, then there exists a 2-round Protocol for Attribute Verification.

Attribute Verification

Theorem 4

If there exists an R3PO for commitment-opening and signature-checking, then there exists a 2-round Protocol for Attribute Verification.

Corollary: Given the following primitives:

- a CP-ABE scheme for general policies
- R3PO for commitment-opening and signature-checking

there exists a Private MA-ABE scheme for general policies.

In Conclusion

- R3PO: Obfuscation of interactively sampled programs
- A library of R3PO instantiations from standard assumptions:
  - Commitment-Opening
  - Signature-Checking
    - Can optionally restrict to a message prefix.
  - Hash-Checking
- A composition theorem to build R3PO for larger program classes.
  - Encapsulates Garbled Circuit Chaining technique
- As an application, we construct Private MA-ABE
- **Open Directions:** More applications, capturing more constructions (e.g., Garbled RAM), adding more features (e.g., blindness)

Thank You

References I

[ABG⁺13] Prabhanjan Ananth, Dan Boneh, Sanjam Garg, Amit Sahai, and Mark Zhandry, Differing-inputs obfuscation and applications, Cryptology ePrint Archive, Paper 2013/689, 2013, https://eprint.iacr.org/2013/689.

[BBC⁺13] Boaz Barak, Nir Bitansky, Ran Canetti, Yael Tauman Kalai, Omer Paneth, and Amit Sahai, Obfuscation for evasive functions, Cryptology ePrint Archive, Paper 2013/668, 2013, https://eprint.iacr.org/2013/668.

[BCTKP14] Nir Bitansky, Ran Canetti, Yael Tauman-Kalai, and Omer Paneth, On virtual grey box obfuscation for general circuits, Cryptology ePrint Archive, Paper 2014/554, 2014, https://eprint.iacr.org/2014/554.

[BGI⁺01] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang, On the (im)possibility of obfuscating programs, Cryptology ePrint Archive, Paper 2001/069, 2001, https://eprint.iacr.org/2001/069.

[BL17] Fabrice Benhamouda and Huijia Lin, k-round mpc from k-round ot via garbled interactive circuits, Cryptology ePrint Archive, Paper 2017/1125, 2017, https://eprint.iacr.org/2017/1125.

[CDG⁺17] Chongwon Cho, Nico Döttling, Sanjam Garg, Divya Gupta, Peihan Miao, and Antigoni Polychroniadou, Laconic oblivious transfer and its applications, Cryptology ePrint Archive, Paper 2017/491, 2017, https://eprint.iacr.org/2017/491.

[Cha07] Melissa Chase, Multi-authority attribute based encryption, Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings, Lecture Notes in Computer Science, vol. 4392, Springer, 2007, pp. 515–534.

[DG17a] Nico Döttling and Sanjam Garg, From selective ibe to full ibe and selective hibe, Cryptology ePrint Archive, Paper 2017/957, 2017, https://eprint.iacr.org/2017/957.

[DG17b] ______, Identity-based encryption from the diffie-hellman assumption, Cryptology ePrint Archive, Paper 2017/543, 2017, https://eprint.iacr.org/2017/543.

[DKW20] Pratish Datta, Ilan Komargodski, and Brent Waters, Decentralized multi-authority abe for dnfs from lwe, Cryptology ePrint Archive, Paper 2020/1386, 2020, https://eprint.iacr.org/2020/1386.

[GS17] Sanjam Garg and Akshayaram Srinivasan, Two-round multiparty secure computation from minimal assumptions, Cryptology ePrint Archive, Paper 2017/1156, 2017, https://eprint.iacr.org/2017/1156.

References II

[HRsV07] Susan Hohenberger, Guy N. Rothblum, abhi shelat, and Vinod Vaikuntanathan, Securely obfuscating re-encryption, Theory of Cryptography (Berlin, Heidelberg) (Salil P. Vadhan, ed.), Springer Berlin Heidelberg, 2007, pp. 233–252.

[IPS14] Yuval Ishai, Omkant Pandey, and Amit Sahai, Public-coin differing-inputs obfuscation and its applications, Cryptology ePrint Archive, Paper 2014/942, 2014, https://eprint.iacr.org/2014/942.

[JLS20] Aayush Jain, Huijia Lin, and Amit Sahai, Indistinguishability obfuscation from well-founded assumptions, Cryptology ePrint Archive, Paper 2020/1003, 2020, https://eprint.iacr.org/2020/1003.

[LW10] Allison Lewko and Brent Waters, Decentralizing attribute-based encryption, Cryptology ePrint Archive, Paper 2010/351, 2010, https://eprint.iacr.org/2010/351.

[WZ17] Daniel Wichs and Giorgos Zirdelis, Obfuscating compute-and-compare programs under lwe, Cryptology ePrint Archive, Paper 2017/276, 2017, https://eprint.iacr.org/2017/276.