Fraunhofer FOKUS

Fraunhofer Institute for Open Communication Systems

Structural Testing with Homomorphic Encryption

Andrei Aleksandrov

Agenda

1. Introduction
2. Mathematical Background
3. Structural Testing Protocol: Definition and Guarantees
4. Conclusion

See the paper for proofs, a specialization to real-valued polynomials, and extended discussion on applications/future work!

Part 01

Introduction

Homomorphic Encryption

Homomorphic encryption allows computations over encrypted values – the result is also encrypted

Encryption function is an isomorphism:

$$encrypt(x_1) + encrypt(x_2) = encrypt(x_1 + x_2)$$

The decryption (~inverse) can be evaluated only with a private key

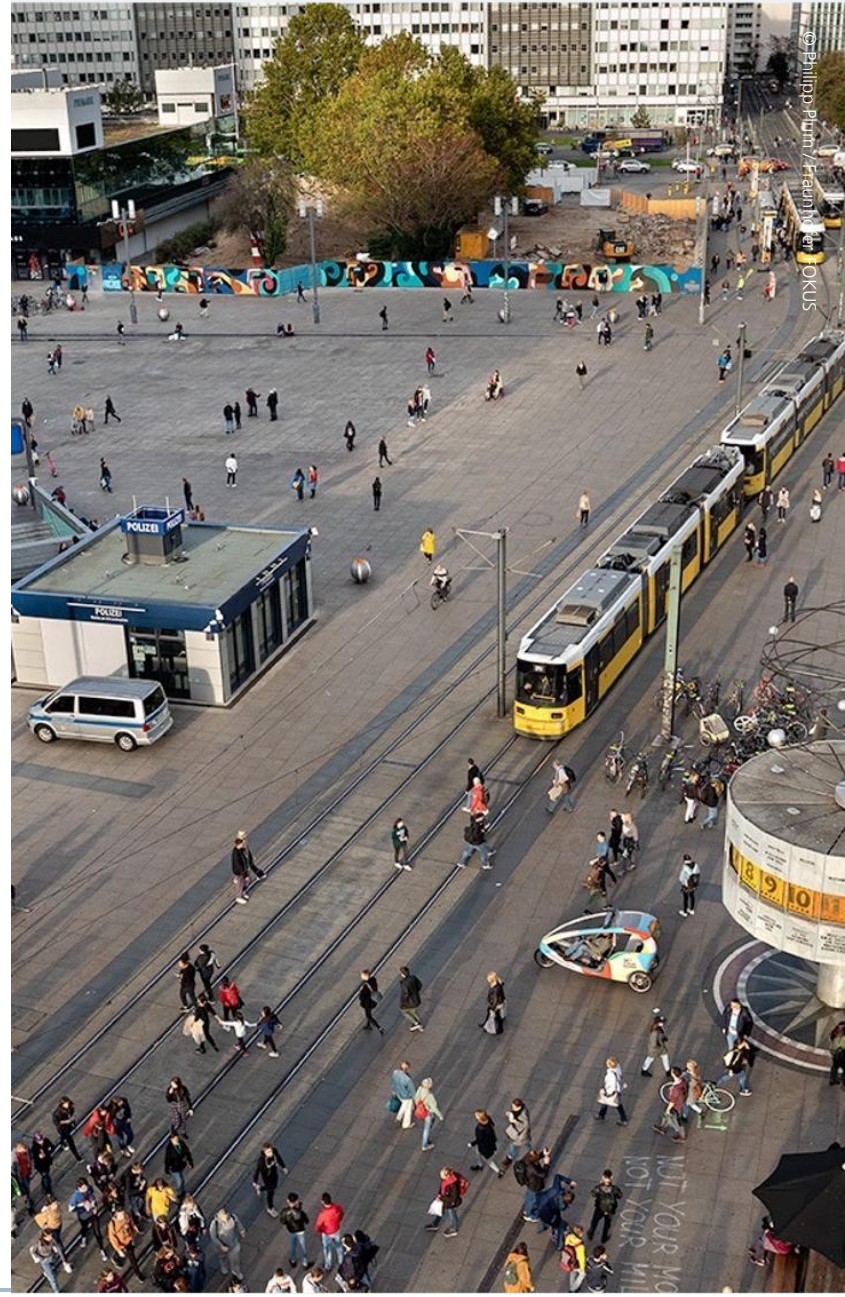
1. Partially homomorphic encryption: Allows for a limited number of operators (e.g. addition only)
2. Somewhat and levelled homomorphic encryption: allows for a larger number of operators with limited number of operations
3. Fully homomorphic encryption: any computable function

Structural Testing with Homomorphic Encryption

1. **Assumption 1:** there is a homomorphic encryption for any algebra with binary operations.

2. **Assumption 2:** every homomorphic encryption only allows operations it was designed for and fails otherwise (non-collision).

3. **Result:** there is a protocol that checks in a black-box setting: given a program P, is P constructed out of operations of a given algebra A?



Structural Testing

Structural specification
A set of algebraic operations a program under test is permitted to use

Structural testing
Checking whether a program satisfies structural specifications (uses permitted operations only)

This work
A black-box structural testing protocol

Motivation for Black Box Structural Testing

1. Enhances black-box testing with new guarantees:

- Checking not only “what” a program does, but also “how”
- Potential to reduce test set sizes using deductive reasoning

2. Complements property testing and polynomial learning:

- Probabilistic results often rely on algebraic assumptions

Theorem 1 (Schwartz, Zippel). *Let*

$$P \in R[x_1, x_2, \ldots, x_n]$$

be a non-zero polynomial of total degree $d \geq 0$ over an integral domain R. Let S be a finite subset of R and let $r_1, r_2, ..., r_n$ be selected at random independently and uniformly from S. Then

$$\Pr[P(r_1, r_2, \ldots, r_n) = 0] \leq \frac{d}{|S|}.$$

Schwartz–Zippel lemma - Wikipedia

THEOREM 1.3. *For every prime power q, there exist constants $\epsilon_1, \epsilon_2 > 0$ such that for every d and n and every function $f : \mathbb{F}_q^n \to \mathbb{F}_q$, it is the case that $\rho_d(f, t_{d,q}) \geq \min\{\epsilon_1 q^{t_{d,q}} \delta(f), \epsilon_2\}$. In other words, the $t_{q,d}$-dimensional test rejects f with probability $\min\{\epsilon_1 q^{t_{q,d}} \delta(f), \epsilon_2\}$, where $t_{q,d}$ is the testing dimension for degree d polynomials over $\mathbb{F}_q$.*

Haramaty, E., Shpilka, A., & Sudan, M. (2013). Optimal testing of multivariate polynomials over small prime fields. *SIAM Journal on Computing*, *42*(2), 536-562.

Part 02

Mathematical Background

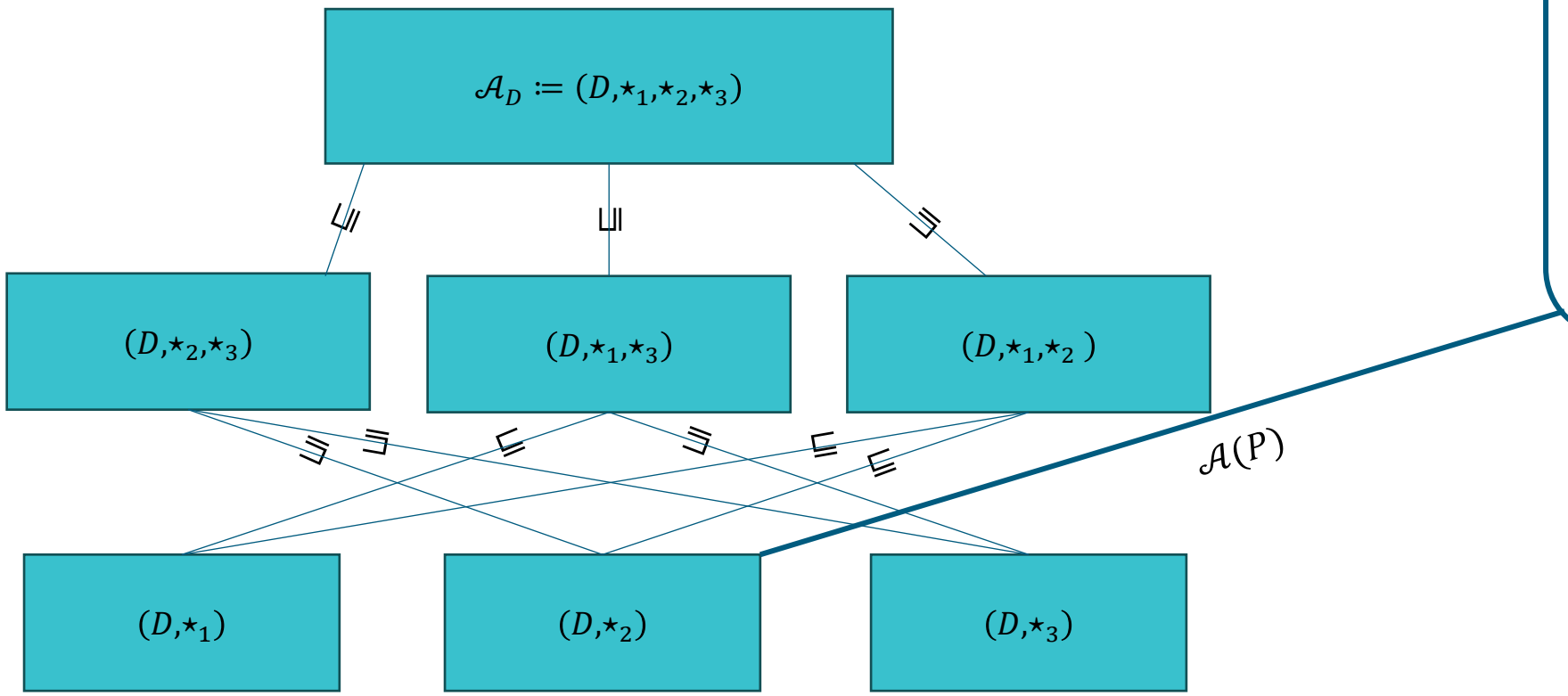
Algebras and Programs

- **Base Algebra:** $\mathcal{A}_D := (D, \star_1, \star_2, \ldots, \star_b)$ with b binary operations $\star_i : D \times D \to D$
- **Set of Operations:** $I_{\mathcal{A}}$, e.g., $I_{\mathcal{A}_D} = \{\star_1, \star_2, \ldots, \star_b\}$
- **Algebraic Program:** a function $P : D^n \to D$ with input variables $x_1, x_2, \ldots, x_n \in D$ represented by a finite recurrence of depth $J > n$:

$$
\begin{aligned}
&P^1 = x_1, P^2 = x_2, \ldots, P^n = x_n \\
&P^j = P^{j_1} \star_j P^{j_2} \qquad n < j < J, j_1 < j, j_2 < j, \star_j \in I_{\mathcal{A}_D} \\
&P(x) = P^J
\end{aligned}
$$

- **Necessary Operation Set:** $\mathcal{P}(P) \subseteq I_{\mathcal{A}_D}$ of a program P is the minimal subset of $I_{\mathcal{A}_D}$ such that for every program term $P^j = P^{j_1} \star_j P^{j_2}$ holds that $\star_j \in \mathcal{P}(P)$.
- **Reduced Algebra:** an algebra A_r is a reduced algebra of A if $I_{A_r} \subseteq I_A$. The comparison of two algebras $A_r \sqsubseteq A$ is true if and only if there is an algebra isomorphic to A_r that is a reduced algebra of A.
- **Necessary Algebra:** for a program P, the necessary algebra $\mathcal{A}(P)$ is a reduced algebra of $\mathcal{A}_D$ such that $I_{\mathcal{A}(P)} = \mathcal{P}(P)$.

Algebras and Programs



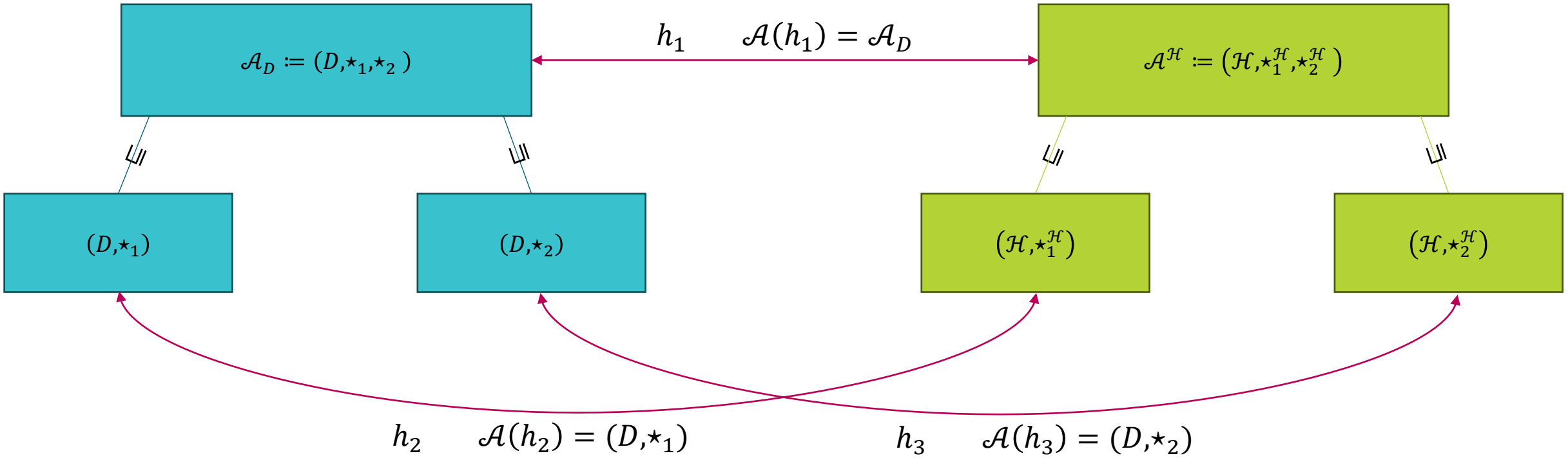
- Program P

$$P^1 = x_1$$
$$P^2 = x_2$$
$$P(x) = P^3 = x_1 \star_2 x_2$$

Homomorphic Encryption and Lifting

- **Encryption Domain:** a set of all possible ciphertexts $\mathcal{H}$.
- **Encrypted Algebra:** an algebra $\mathcal{A}^{\mathcal{H}} \coloneqq \left(\mathcal{H}, \star_1^{\mathcal{H}}, \star_2^{\mathcal{H}}, \ldots, \star_b^{\mathcal{H}}\right)$ isomorphic to $\mathcal{A}_D$.
- **Homomorphic Encryption:** given a $\mathcal{A}_r \sqsubseteq \mathcal{A}_D$, there exists an isomorphic $\mathcal{A}_r^{\mathcal{H}} \sqsubseteq \mathcal{A}^{\mathcal{H}}$. A homomorphic encryption $h: D \to \mathcal{H}$ is an isomorphism between $\mathcal{A}_r$ and $\mathcal{A}_r^{\mathcal{H}}$. We define the supported algebra of h as $\mathcal{A}(h) \coloneqq \mathcal{A}_r$.

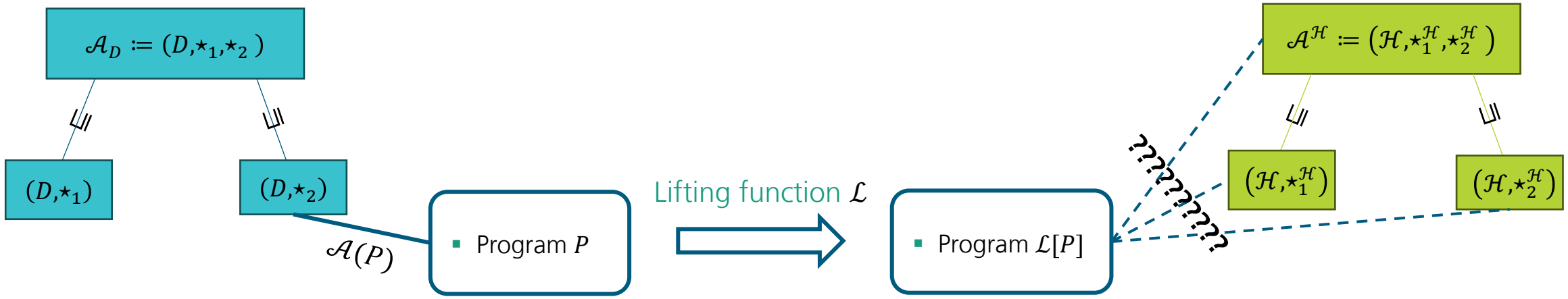
Homomorphic Encryption and Lifting



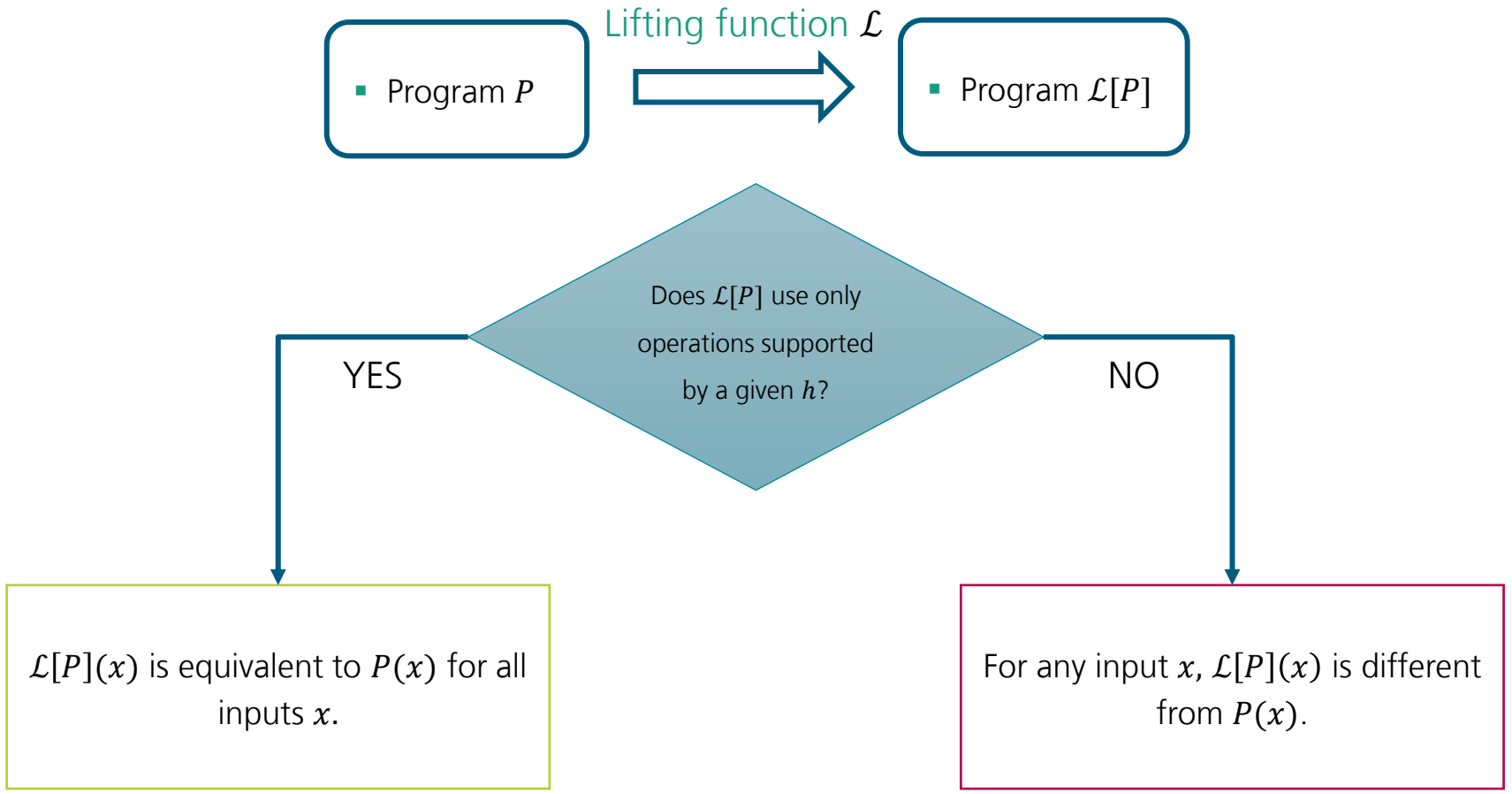
Homomorphic Encryption and Lifting

- **Encryption Domain:** a set of all possible ciphertexts $\mathcal{H}$.
- **Encrypted Algebra:** an algebra $\mathcal{A}^{\mathcal{H}} \coloneqq \left(\mathcal{H}, \star_1^{\mathcal{H}}, \star_2^{\mathcal{H}}, \ldots, \star_b^{\mathcal{H}}\right)$ isomorphic to $\mathcal{A}_D$.
- **Homomorphic Encryption:** given a $\mathcal{A}_r \sqsubseteq \mathcal{A}_D$, there exists an isomorphic $\mathcal{A}_r^{\mathcal{H}} \sqsubseteq \mathcal{A}^{\mathcal{H}}$. A homomorphic encryption $h: D \to \mathcal{H}$ is an isomorphism between $\mathcal{A}_r$ and $\mathcal{A}_r^{\mathcal{H}}$. We define the supported algebra of h as $\mathcal{A}(h) \coloneqq \mathcal{A}_r$.
- **Lifting:** let $\mathbb{P}_{\mathcal{A}}$ be a space of all possible programs over $\mathcal{A}$. A lifting function is a total function $\mathcal{L}: \mathbb{P}_{\mathcal{A}_D} \to \mathbb{P}_{\mathcal{A}^{\mathcal{H}}}$. We call the result $\mathcal{L}[P]$ a lifting of P.

Homomorphic Encryption and Lifting



Non-Collision Assumption



Homomorphic Encryption and Lifting

- **Encryption Domain:** a set of all possible ciphertexts $\mathcal{H}$.
- **Encrypted Algebra:** an algebra $\mathcal{A}^{\mathcal{H}} \coloneqq \left(\mathcal{H}, \star_1^{\mathcal{H}}, \star_2^{\mathcal{H}}, \ldots, \star_b^{\mathcal{H}}\right)$ isomorphic to $\mathcal{A}_D$.
- **Homomorphic Encryption:** given a $\mathcal{A}_r \sqsubseteq \mathcal{A}_D$, there exists an isomorphic $\mathcal{A}_r^{\mathcal{H}} \sqsubseteq \mathcal{A}^{\mathcal{H}}$. A homomorphic encryption $h: D \to \mathcal{H}$ is an isomorphism between $\mathcal{A}_r$ and $\mathcal{A}_r^{\mathcal{H}}$. We define the supported algebra of h as $\mathcal{A}(h) \coloneqq \mathcal{A}_r$.
- **Lifting:** let $\mathbb{P}_{\mathcal{A}}$ be a space of all possible programs over $\mathcal{A}$. A lifting function is a total function $\mathcal{L}: \mathbb{P}_{\mathcal{A}_D} \to \mathbb{P}_{\mathcal{A}^{\mathcal{H}}}$. We call the result $\mathcal{L}[P]$ a lifting of P.
- **Non-Collision Assumption:** for all lifting functions $\mathcal{L}$, all $P \in \mathbb{P}_{\mathcal{A}_D}$, and all homomorphic encryptions h, the following two hold:
  1. $\mathcal{A}(\mathcal{L}[P]) \sqsubseteq \mathcal{A}(h)$ if and only if for all $x \in D^n, h^{-1}\left(\mathcal{L}[P]\left(h(x)\right)\right) = P(x)$,
  2. $\mathcal{A}(\mathcal{L}[P]) \not\sqsubseteq \mathcal{A}(h)$ if and only if for all $x \in D^n, h^{-1}\left(\mathcal{L}[P]\left(h(x)\right)\right) \neq P(x)$.

Part 03

Structural Testing Protocol: Definition and Guarantees

Black-Box Structural Testing Protocol

Agent A (the client)

Agent B (the service)

Black-Box Structural Testing Protocol

Agent A (the client)

Step 1
The client securely generates a test set

Agent B (the service)

Test set $\mathcal{X} \subseteq D^n$

Program P_A
(test oracle)

Program P_B
(under test)

Black-Box Structural Testing Protocol

Agent A (the client)

Step 2
The client selects a homomorphic encryption and notifies the service about it

Agent B (the service)

Test set $\mathcal{X} \subseteq D^n$

Program P_A
(test oracle)

Encryption h with
$\mathcal{A}(h) = \mathcal{A}(P_A)$

Notification

Program P_B
(under test)

Black-Box Structural Testing Protocol

Agent A (the client)

Test set $\mathcal{X} \subseteq D^n$

Program P_A
(test oracle)

Encryption h with
$\mathcal{A}(h) = \mathcal{A}(P_A)$

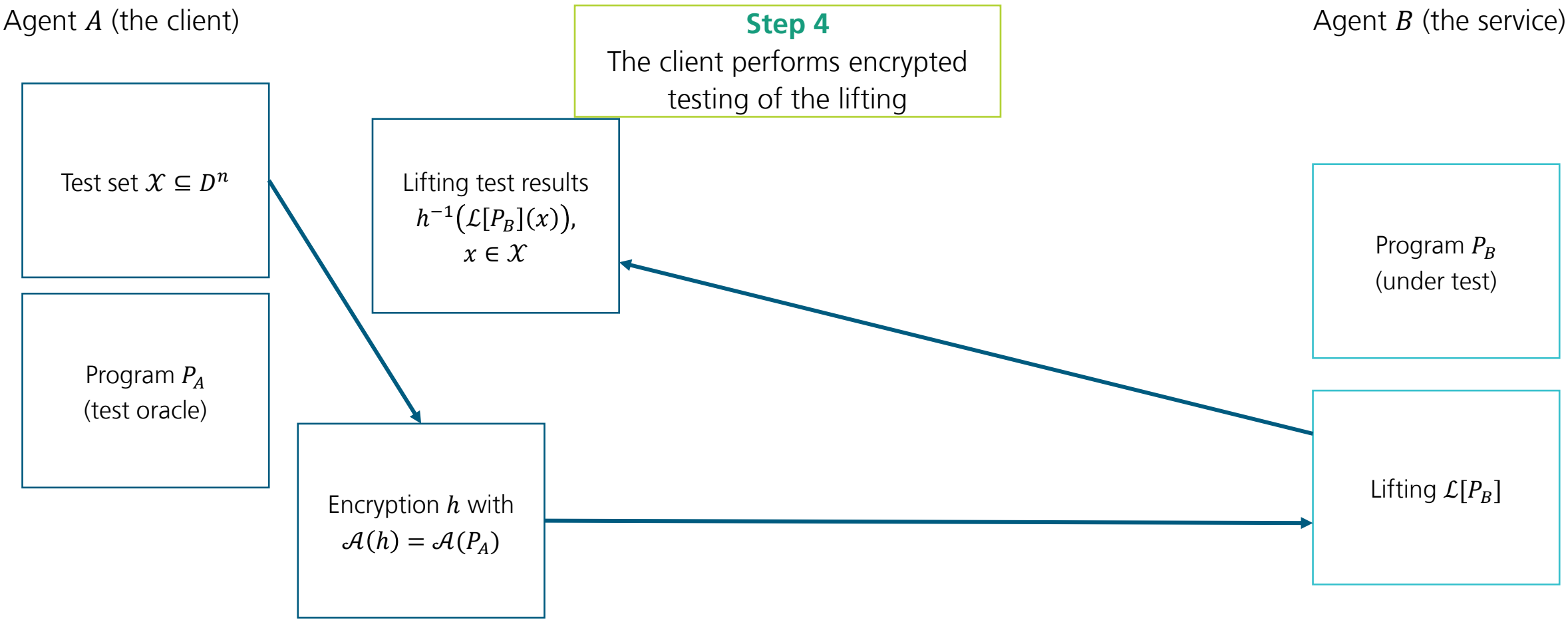
Step 3
The service constructs a lifting of program under test

Agent B (the service)

Program P_B
(under test)

Lifting $\mathcal{L}[P_B]$

Black-Box Structural Testing Protocol

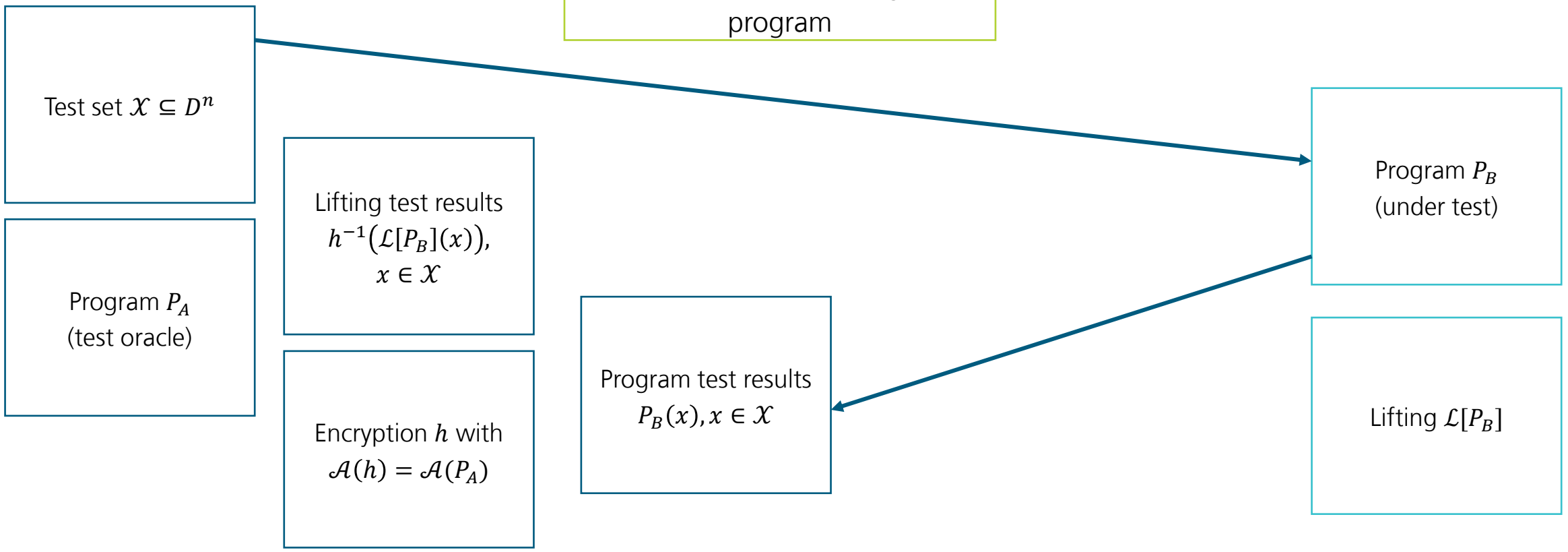


Black-Box Structural Testing Protocol

Agent A (the client)

Step 5
The client performs testing of the program

Agent B (the service)



Black-Box Structural Testing Protocol

Agent A (the client)

Test set $\mathcal{X} \subseteq D^n$

Lifting test results
$h^{-1}\big(\mathcal{L}[P_B](x)\big)$,
$x \in \mathcal{X}$

=

Program P_A
(test oracle)

=

Encryption h with
$\mathcal{A}(h) = \mathcal{A}(P_A)$

Program test results
$P_B(x), x \in \mathcal{X}$

Step 6

The client compares both test results with the test oracle, returns true ⊤ if all three equal, else false ⊥

Agent B (the service)

Program P_B
(under test)

Lifting $\mathcal{L}[P_B]$

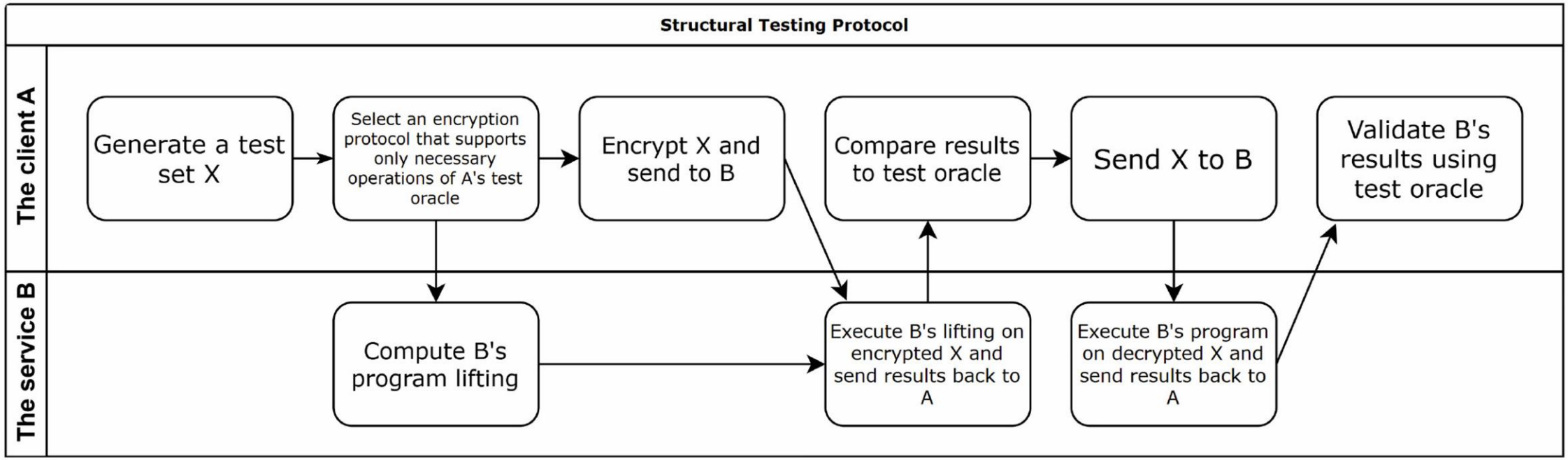
Protocol Guarantee

Theorem 1 (Protocol Correctness). For all programs $P_A, P_B : D^n \to D$, after an execution of the black-box structural testing protocol, the following two statements are equivalent:

1. The protocol returns $\top$.

2. $\mathcal{A}(\mathcal{L}[P_B]) \sqsubseteq \mathcal{A}(P_A)$ and for all $x \in \mathcal{X}, P_A(x) = P_B(x) = h^{-1}(\mathcal{L}[P_B](h(x)))$.

Black-Box Structural Testing Protocol



Part 04

Conclusion

Conclusion

Summary

- Structural testing – checking that a program uses only permitted operations.
- Main result: a protocol for black-box structural testing.
- Limitations: mathematical assumptions, implementation hurdles.

Future work

- Design of encryption protocols supporting the assumptions.
- Identification of useful algebras for structural testing.
- Developing algorithms for liftings.
- Possible impact: turning black-box into gray-box, reducing test effort through deduction.

Structural Testing with HE

We did prove that there is a theoretical possibility to extend black-box testing with a structural specification – a set of permitted operations

Thank you for your attention!

Research Scientist
Andrei Aleksandrov
andrei.aleksandrov@fokus.fraunhofer.de

Fraunhofer Institute for
Open Communication Systems FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany
info@fokus.fraunhofer.de
www.fokus.fraunhofer.de