Categories for Cryptographic Composability

Riley Shahar
Advised by Angélica Osorno and Adam Groce

Outline

Outline

- Cryptographic composability

Outline

- Cryptographic composability
- Why categories?

Outline

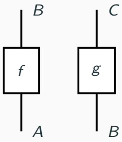
- Cryptographic composability
- Why categories?
- Towards a categorical theory of cryptography

Outline

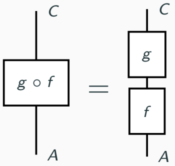
- Cryptographic composability
- Why categories?
- Towards a categorical theory of cryptography
- Open problems

Composition

Composition

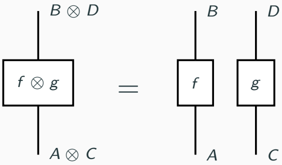


Composition



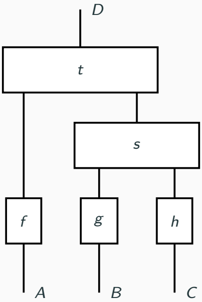
Sequential (Vertical) Composition

Composition



Parallel (Horizontal) Composition

Composition



Cryptography

Cryptography is *the mathematics of secure computation.*

Cryptography

Cryptography is *the mathematics of secure computation.*

Cryptography

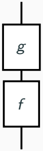
Cryptography is *the mathematics of secure computation.*

Cryptographic Composability

Say f and g are secure under some definition.

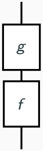
Cryptographic Composability

Say f and g are secure under some definition.

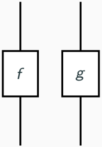
Is  secure?

Cryptographic Composability

Say f and g are secure under some definition.

Is 

secure?

Is 

secure?

Simulation-based security

Simulation-based security

Compare the protocol to an *ideal world* with a trusted third party.

Simulation-based security

Compare the protocol to an *ideal world* with a trusted third party.

A protocol is *secure* if it is computationally indistinguishable from the ideal world.

Security Parameters

Cryptosystems are parametrized by a *security parameter* $n \in \mathbb{N}$.

Security Parameters

Cryptosystems are parametrized by a *security parameter* $n \in \mathbb{N}$.

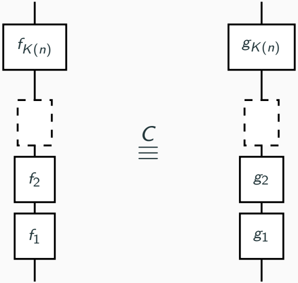
Let $f_i \stackrel{C}{\equiv} g_i$.

Security Parameters

Cryptosystems are parametrized by a *security parameter* $n \in \mathbb{N}$.

Let $f_i \overset{C}{\equiv} g_i$.

Does



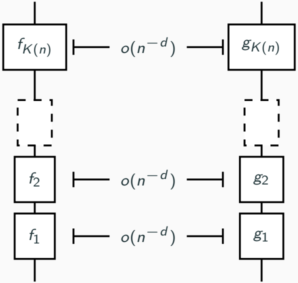
hold?

Security Parameters

Cryptosystems are parametrized by a *security parameter* $n \in \mathbb{N}$.

Let $f_i \overset{C}{\equiv} g_i$.

Does



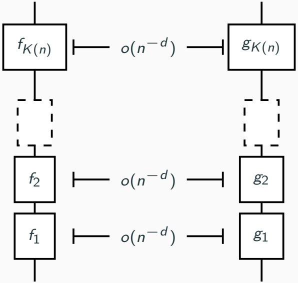
hold?

Security Parameters

Cryptosystems are parametrized by a *security parameter* $n \in \mathbb{N}$.

Let $f_i \stackrel{C}{\equiv} g_i$.

Does



hold?   Only if $K(n) = O(n^d)$.

Simulation-based composability

Theorem [MR92]. *Simulation-secure protocols compose securely in sequences of polynomial length.*

Simulation-based composability

Theorem [MR92]. *Simulation-secure protocols compose securely in sequences of polynomial length.*

However, [GK96] gave a protocol that's simulation secure, but doesn't compose in parallel.

Universal Composability

Universal Composability

Canetti (2000) defined the notion of *UC-security*.

Universal Composability

Canetti (2000) defined the notion of *UC-security*.

Theorem [Can00]. *UC-secure protocols compose securely in parallel sequences of polynomial length or width.*

Universal Composability

Canetti (2000) defined the notion of *UC-security*.

Theorem [Can00]. *UC-secure protocols compose securely in parallel sequences of polynomial length or width.*

UC was revised 9 times so far, most recently in 2020.

Universal Composability

Canetti (2000) defined the notion of *UC-security*.

Theorem [Can00]. *UC-secure protocols compose securely in parallel sequences of polynomial length or width.*

UC was revised 9 times so far, most recently in 2020.

The proofs...

- are dependent on technical details;

Universal Composability

Canetti (2000) defined the notion of *UC-security*.

Theorem [Can00]. *UC-secure protocols compose securely in parallel sequences of polynomial length or width.*

UC was revised 9 times so far, most recently in 2020.

The proofs...

- are dependent on technical details;
- leave artifacts in the protocol;

Universal Composability

Canetti (2000) defined the notion of *UC-security*.

Theorem [Can00]. *UC-secure protocols compose securely in parallel sequences of polynomial length or width.*

UC was revised 9 times so far, most recently in 2020.

The proofs...

- are dependent on technical details;
- leave artifacts in the protocol;
- are hard to trust.

Cryptography is in need of an elegant mathematical theory abstracting composability of computational processes. . .

Cryptography is in need of an elegant mathematical theory abstracting composability of computational processes...

...**category theory** is an excellent candidate for such a theory.

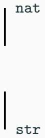
Categories and Programming Languages

In programming language theory:

Categories and Programming Languages

In programming language theory:

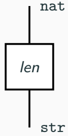
- objects are types;



Categories and Programming Languages

In programming language theory:

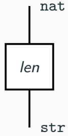
- objects are types;
- morphisms are programs.



Categories and Programming Languages

In programming language theory:

- objects are types;
- morphisms are programs.



(We've already been doing category theory!)

Category Theory and Computer Science

Category theory has been applied to:

Category Theory and Computer Science

Category theory has been applied to:

- databases [BSW94];

Category Theory and Computer Science

Category theory has been applied to:

- databases [BSW94];
- architecture [HC01];

Category Theory and Computer Science

Category theory has been applied to:

- databases [BSW94];
- architecture [HC01];
- machine learning [Cru+21];

Category Theory and Computer Science

Category theory has been applied to:

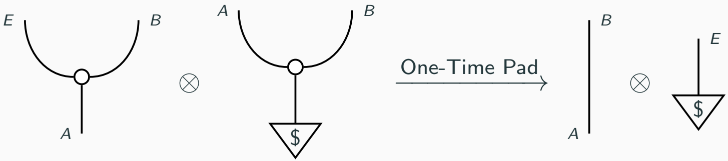
- databases [BSW94];
- architecture [HC01];
- machine learning [Cru+21];
- robotics [Agu+23];

Category Theory and Computer Science

Category theory has been applied to:

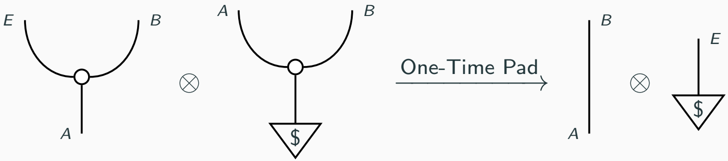
- databases [BSW94];
- architecture [HC01];
- machine learning [Cru+21];
- robotics [Agu+23];
- cryptography [BK22]!

Categorical Composable Cryptography: The Idea



[Broadbent & Karvonen, 2022]

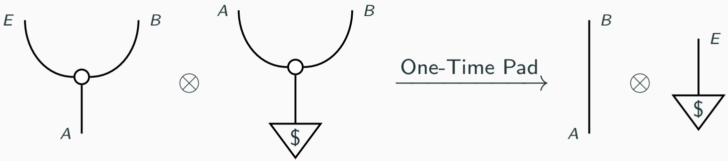
Categorical Composable Cryptography: The Idea



[Broadbent & Karvonen, 2022]

- objects are resources, like channels or keys;

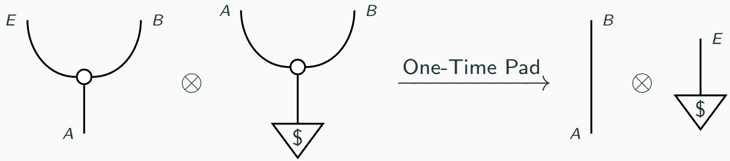
Categorical Composable Cryptography: The Idea



[Broadbent & Karvonen, 2022]

- objects are resources, like channels or keys;
- morphisms are "protocols with holes";

Categorical Composable Cryptography: The Idea



[Broadbent & Karvonen, 2022]

- objects are resources, like channels or keys;
- morphisms are "protocols with holes";
- composition "plugs in the holes".

Advantages of Categorical Composable Cryptography

Categorical composable cryptography:

Advantages of Categorical Composable Cryptography

Categorical composable cryptography:

- obtains a general composition theorem;

Advantages of Categorical Composable Cryptography

Categorical composable cryptography:

- obtains a general composition theorem;
- abstracts over all SMCs;

Advantages of Categorical Composable Cryptography

Categorical composable cryptography:

- obtains a general composition theorem;
- abstracts over all SMCs;
- abstracts over adversarial capabilities;

Advantages of Categorical Composable Cryptography

Categorical composable cryptography:

- obtains a general composition theorem;
- abstracts over all SMCs;
- abstracts over adversarial capabilities;
- abstracts over adversarial goals;

Advantages of Categorical Composable Cryptography

Categorical composable cryptography:

- obtains a general composition theorem;
- abstracts over all SMCs;
- abstracts over adversarial capabilities;
- abstracts over adversarial goals;
- proofs are pictoral and “straightforward”.

Advantages of Categorical Composable Cryptography

Categorical composable cryptography:

- obtains a general composition theorem;
- abstracts over all SMCs;
- abstracts over adversarial capabilities;
- abstracts over adversarial goals;
- proofs are pictoral and “straightforward”.

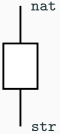
There’s always a tradeoff: it relies on dense abstract machinery.

Adversaries Break Things

Adversaries can violate the type system.

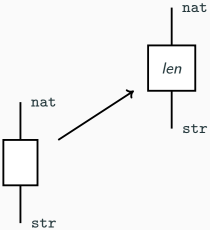
Adversaries Break Things

Adversaries can violate the type system.



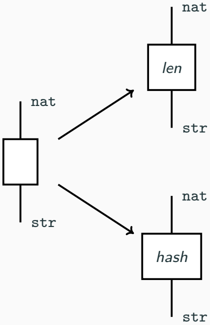
Adversaries Break Things

Adversaries can violate the type system.



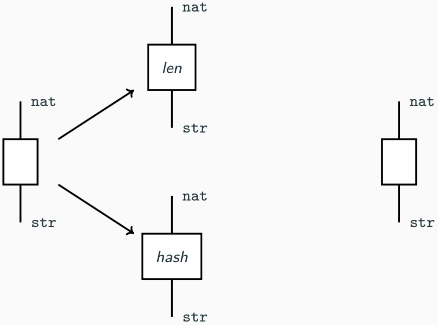
Adversaries Break Things

Adversaries can violate the type system.



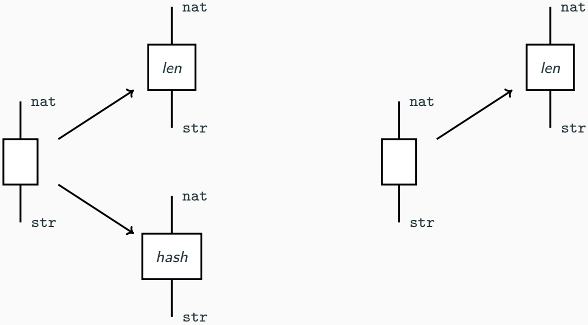
Adversaries Break Things

Adversaries can violate the type system.



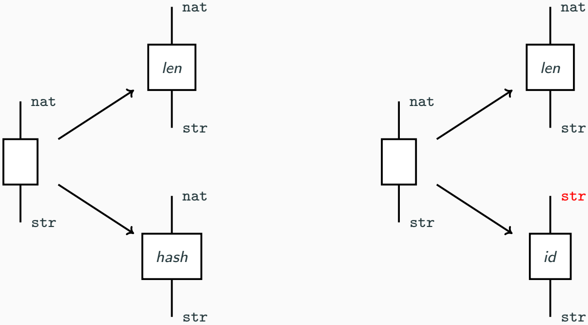
Adversaries Break Things

Adversaries can violate the type system.



Adversaries Break Things

Adversaries can violate the type system.



Attack Models

In CCC, adversaries are constrained by *attack models*.

Attack Models

In CCC, adversaries are constrained by *attack models*.

Open Question 1: *Can the axioms be formulated as functoriality plus some conditions?*

Attack Models

In CCC, adversaries are constrained by *attack models*.

~~Open~~ Question 1: *Can the axioms be formulated as functoriality plus some conditions?* Answer: **Yes.**

Attack Models

In CCC, adversaries are constrained by *attack models*.

~~Open~~ Question 1: *Can the axioms be formulated as functoriality plus some conditions?* Answer: **Yes.**

Open Question 2: *How broad is the definition of an attack model? Does it capture enough of modern cryptography?*

Computational Indistinguishability

Composition should only work polynomially many times.

Computational Indistinguishability

Composition should only work polynomially many times. In fact, computational indistinguishability is not even an equivalence relation.

Computational Indistinguishability

Composition should only work polynomially many times. In fact, computational indistinguishability is not even an equivalence relation.

B&K work around this by artificially limiting the universe size.

Computational Indistinguishability

Composition should only work polynomially many times. In fact, computational indistinguishability is not even an equivalence relation.

B&K work around this by artificially limiting the universe size.

Open Question 3: *Is there a natural categorical model of computational indistinguishability?*

What Happens Next?

What Happens Next?

- Can we incorporate categorical notions from game theory, programming languages, etc. into cryptography?

What Happens Next?

- Can we incorporate categorical notions from game theory, programming languages, etc. into cryptography?
- What does the presence of various categorical structure ((co)limits, monads, etc.) say cryptographically?

References

Esther Aguado et al. **Category Theory for Autonomous Robots: The Marathon 2 Use Case.** 2023. arXiv: 2303.01152 [cs.RO].

Anne Broadbent and Martti Karvonen. **"Categorical composable cryptography".** In: *Foundations of software science and computation structures.* Vol. 13242. Lecture Notes in Comput. Sci. Springer, Cham, 2022, pp. 161–183. ISBN: 9783030992538. DOI: 10.1007/978-3-030-99253-8_9. URL: https://doi.org/10.1007/978-3-030-99253-8_9.

Kenneth Baclawski, Dan Simovici, and William White. **"A categorical approach to database semantics".** In: *Mathematical Structures in Computer Science* 4.2 (1994), pp. 147–183. DOI: 10.1017/S0960129500000426.

Ran Canetti. **Universally Composable Security: A New Paradigm for Cryptographic Protocols.** Cryptology ePrint Archive, Paper 2000/067. https://eprint.iacr.org/2000/067. 2000. URL: https://eprint.iacr.org/2000/067.

G. S. H. Cruttwell et al. **Categorical Foundations of Gradient-Based Learning.** 2021. arXiv: 2103.01931 [cs.LG].

Oded Goldreich and Hugo Krawczyk. **"On the Composition of Zero-Knowledge Proof Systems".** In: *SIAM Journal on Computing* 25.1 (1996), pp. 169–192. DOI: 10.1137/S0097539791220688. eprint: https://doi.org/10.1137/S0097539791220688. URL: https://doi.org/10.1137/S0097539791220688.

M.J. Healy and T.P. Caudell. **"A categorical semantic analysis of ART architectures".** In: *IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No.01CH37222).* Vol. 1. 2001, 38–43 vol.1. DOI: 10.1109/IJCNN.2001.938988.

Silvio Micali and Phillip Rogaway. **"Secure Computation".** In: *Advances in Cryptology — CRYPTO '91.* Ed. by Joan Feigenbaum. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, pp. 392–404. ISBN: 978-3-540-46766-3.