What's going on with categorical composable cryptography?

Martti Karvonen

University College London

SYCO
April 25 2025

Structure of the talk

- Part 1: Categorical composable cryptography
- Part 2: Capturing other frameworks
- Part 3: On game-based (not necessarily composable) cryptography

Part 1: Categorical composable cryptography

Anne Broadbent [1] Martti Karvonen[2]

[1]University of Ottawa

[2]University College London

Overview

- motivation: standard cryptography is not composable. Existing approaches to make it composable are a bit hacky/tedious/very complicated and seem to beg a categorical formalization

Overview

- motivation: standard cryptography is not composable. Existing approaches to make it composable are a bit hacky/tedious/very complicated and seem to beg a categorical formalization
- main idea: cryptography as a resource theory — the resources are various functionalities (e.g. keys, channels etc) and transformations are given by protocols that build the target resource *securely* from the starting resources.

Overview

- motivation: standard cryptography is not composable. Existing approaches to make it composable are a bit hacky/tedious/very complicated and seem to beg a categorical formalization
- main idea: cryptography as a resource theory — the resources are various functionalities (e.g. keys, channels etc) and transformations are given by protocols that build the target resource *securely* from the starting resources.
  - categories of correct resource conversions as a Grothendieck construction

Overview

- motivation: standard cryptography is not composable. Existing approaches to make it composable are a bit hacky/tedious/very complicated and seem to beg a categorical formalization
- main idea: cryptography as a resource theory — the resources are various functionalities (e.g. keys, channels etc) and transformations are given by protocols that build the target resource *securely* from the starting resources.
  - categories of correct resource conversions as a Grothendieck construction
  - correct and *secure* conversions as a subcategory

Overview

- motivation: standard cryptography is not composable. Existing approaches to make it composable are a bit hacky/tedious/very complicated and seem to beg a categorical formalization
- main idea: cryptography as a resource theory — the resources are various functionalities (e.g. keys, channels etc) and transformations are given by protocols that build the target resource *securely* from the starting resources.
  - categories of correct resource conversions as a Grothendieck construction
  - correct and *secure* conversions as a subcategory
- example(ish): one-time-pad (OTP) as a transformation
  OTP: $key \otimes insecure\ channel \to secure\ channel$
  Security & correctness of OTP boil down to axioms of a Hopf algebra with an integral.

Real-world ideal-world paradigm

AKA simulation paradigm. Standard meta-approach for composable security.

Usual definition: a real protocol P securely realizes the ideal functionality F from the resource R if for any attack A on $P \circ R$ there is a simulator S on F such that $(A, P) \circ R$ is indistinguishable from $S \circ F$ by any (efficient) environment.

"Any bad thing that could happen during the protocol could also happen in the ideal world."

Real-world ideal-world paradigm

AKA simulation paradigm. Standard meta-approach for composable security.

Usual definition: a real protocol P securely realizes the ideal functionality F from the resource R if for any attack A on $P \circ R$ there is a simulator S on F such that $(A, P) \circ R$ is indistinguishable from $S \circ F$ by any (efficient) environment.

"Any bad thing that could happen during the protocol could also happen in the ideal world."

Usual ways of making this precise:

- Fixing a concrete low-level formalism for interactive computation (e.g. UC-security)
- Abstract cryptography and constructive cryptography — close to our work in spirit but technically different

N+1th approach

In this work we formalize the simulation paradigm over an arbitrary category (and a model of attacks). The main result is that protocols secure against a fixed attack model can be composed sequentially and in parallel.

N+1th approach

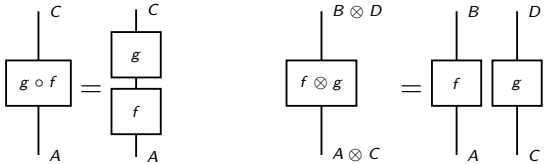
In this work we formalize the simulation paradigm over an arbitrary category (and a model of attacks). The main result is that protocols secure against a fixed attack model can be composed sequentially and in parallel. Some benefits:

- simulation-based security definitions are inherently composable, whether the model of computation is synchronous or not, classical or quantum etc.
- abstract attack models pave way for other kinds of attackers than malicious ones
- different notions of security (computational, finite-key regimen etc) fit in
- benefits of CT: (i) tools, in particular string diagrams (ii) potential connections to other fields

Recap on pictures

Let **C** be a symmetric monoidal category — concretely, you can think of (finite) sets and stochastic maps. We will depict a morphism as 

, and composition and monoidal product as



Special morphisms get nicer pictures: identities and symmetries are

Resource theories

Roughly: An SMC where you mostly care whether a hom-set is empty or not.

Examples:

- Can these noisy channels be used to simulate a (almost) noiseless channel?

Resource theories

Roughly: An SMC where you mostly care whether a hom-set is empty or not.
Examples:

- Can these noisy channels be used to simulate a (almost) noiseless channel?
- Is there a LOCC-protocol that transforms this quantum state to that one?

Resource theories

Roughly: An SMC where you mostly care whether a hom-set is empty or not.
Examples:

- Can these noisy channels be used to simulate a (almost) noiseless channel?
- Is there a LOCC-protocol that transforms this quantum state to that one?
- Any preordered commutative monoid.

Resource theories

Roughly: An SMC where you mostly care whether a hom-set is empty or not.
Examples:

- Can these noisy channels be used to simulate a (almost) noiseless channel?
- Is there a LOCC-protocol that transforms this quantum state to that one?
- Any preordered commutative monoid.

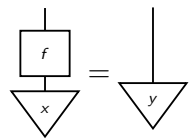
In "A mathematical theory of resources" Coecke, Fritz & Spekkens construct many resource theories starting from an SMC $\mathbf{C}$ equipped with a wide sub-SMC $\mathbf{C}_F$ of free processes, and show how familiar examples are captured by these.

Resource theories

Roughly: An SMC where you mostly care whether a hom-set is empty or not.
Examples:

- Can these noisy channels be used to simulate a (almost) noiseless channel?
- Is there a LOCC-protocol that transforms this quantum state to that one?
- Any preordered commutative monoid.

In "A mathematical theory of resources" Coecke, Fritz & Spekkens construct many resource theories starting from an SMC $\mathbf{C}$ equipped with a wide sub-SMC $\mathbf{C}_F$ of free processes, and show how familiar examples are captured by these. One of the constructions – the resource theory of states – is defined as follows:
Objects are states of $\mathbf{C}$, i.e. maps out of I, and maps $x \to y$ are maps f in $\mathbf{C}_F$ such that



.

Grothendieck construction

This is the Grothendieck construction applied to the composite $\mathbf{C}_F \hookrightarrow \mathbf{C} \xrightarrow{\hom(I,-)} \mathbf{Set}$.

Grothendieck construction

This is the Grothendieck construction applied to the composite $\mathbf{C}_F \hookrightarrow \mathbf{C} \xrightarrow{\hom(I,-)} \mathbf{Set}$.

Recall that, for any functor $F\colon \mathbf{C} \to \mathbf{Set}$ we can build a category $\int F$: its objects are pairs (A, r) with $A \in \mathbf{C}$ and $r \in F(A)$, and maps $(A, r) \to (B, s)$ are given by maps $f\colon A \to B$ in $\mathbf{C}$ such that $F(f)r = s$

Grothendieck construction

This is the Grothendieck construction applied to the composite $\mathbf{C}_F \hookrightarrow \mathbf{C} \xrightarrow{\hom(I,-)} \mathbf{Set}$.

Recall that, for any functor $F\colon \mathbf{C} \to \mathbf{Set}$ we can build a category $\int F$: its objects are pairs (A, r) with $A \in \mathbf{C}$ and $r \in F(A)$, and maps $(A, r) \to (B, s)$ are given by maps $f\colon A \to B$ in $\mathbf{C}$ such that $F(f)r = s$

Whenever F is lax symmetric monoidal, $\int F$ is a symmetric monoidal category, see

'*Monoidal Grothendieck construction*'

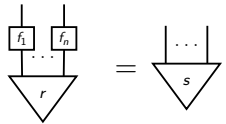
Moeller & Vasilakopoulou, TAC 2020.

Running example: n-partite states and transformations

If $\mathbf{C}$ is symmetric monoidal, let us compute the end result for

$$\mathbf{C}_F^n \hookrightarrow \mathbf{C}^n \xrightarrow{\otimes} \mathbf{C} \xrightarrow{\hom(I,-)} \mathbf{Set}.$$

Its objects are of the form $((A_i)_{i=1}^n, r\colon I \to \bigotimes A_i)$. A map $(((A_i)_{i=1}^n, r) \to (((B_i)_{i=1}^n, s)$ is then a tuple $(f_i)_{i=1}^n$ of morphisms of $\mathbf{C}_F$ that transforms r to s:

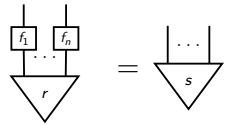


Running example: n-partite states and transformations

If $\mathbf{C}$ is symmetric monoidal, let us compute the end result for

$$\mathbf{C}_F^n \hookrightarrow \mathbf{C}^n \xrightarrow{\otimes} \mathbf{C} \xrightarrow{\hom(I,-)} \mathbf{Set}.$$

Its objects are of the form $((A_i)_{i=1}^n, r\colon I \to \bigotimes A_i)$. A map $(((A_i)_{i=1}^n, r) \to (((B_i)_{i=1}^n, s)$ is then a tuple $(f_i)_{i=1}^n$ of morphisms of $\mathbf{C}_F$ that transforms r to s:



We think of this as a resource theory with n-parties who try to agree on actions $f_1, \dots f_n$ to transform some resource to another one.

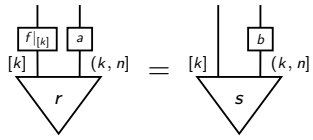
Security in the running example

Such protocols are not necessarily secure—what if some subset of the parties does something else instead?

Security in the running example

Such protocols are not necessarily secure—what if some subset of the parties does something else instead?

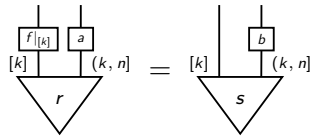
Assume the first k parties are honest and the last $n - k$ parties are dishonest. Then $(f_1, \ldots f_n)$ is secure if for any a there is a b such that



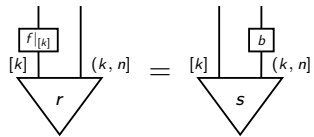
Security in the running example

Such protocols are not necessarily secure—what if some subset of the parties does something else instead?

Assume the first k parties are honest and the last $n-k$ parties are dishonest. Then $(f_1, \ldots f_n)$ is secure if for any a there is a b such that



It suffices to check this for the initial attack $\bigotimes_{k+1}^{n} \mathrm{id}$:



Security in the abstract

Usually a resource theory talks only about correct transformations

To add in security:

- need an attack model $\mathcal{A}$ that gives for each protocol f a collection $\mathcal{A}(f)$ of attacks on it, satisfying some axioms.
- security against $\mathcal{A}$: for each attack on the protocol there is an attack on the target with similar end-results

Security in the abstract

Usually a resource theory talks only about correct transformations

To add in security:

- need an attack model $\mathcal{A}$ that gives for each protocol f a collection $\mathcal{A}(f)$ of attacks on it, satisfying some axioms.
- security against $\mathcal{A}$: for each attack on the protocol there is an attack on the target with similar end-results

Definition

An attack model on $\mathbf{C}$ gives for each f a collection $\mathcal{A}(f)$ of morphisms in $\mathbf{C}$ such that (i) $\mathcal{A}(gf) = \mathcal{A}(g) \circ \mathcal{A}(f)$ and (ii) $\mathcal{A}(g \otimes f) = \mathcal{A}(\mathrm{id}) \circ (\mathcal{A}(g) \otimes \mathcal{A}(f))$ and ...

Note: If $f : A \to B$, we don't require $\mathcal{A}(f) \subset \mathbf{C}(A, B)$ — attackers don't care about our type system!

Security in the abstract II

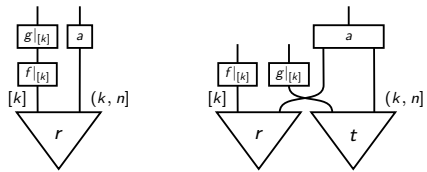
Definition

An attack model on **C** gives for each f a collection $\mathcal{A}(f)$ of morphisms in **C** such that

(i) $\mathcal{A}(g \circ f) = \mathcal{A}(g) \circ \mathcal{A}(f)$ and

(ii) $\mathcal{A}(g \otimes f) = \mathcal{A}(\mathrm{id}) \circ (\mathcal{A}(g) \otimes \mathcal{A}(f))$ and ...

For malicious adversaries we can use identities/wires to get the factorizations



Composability

Theorem

Protocols secure against an attack model $\mathcal{A}$ are closed under composition ($\circ$ and $\otimes$).

Proof.

$\otimes$ and $\circ$ inherited form the ambient category—one just needs to check that they work.

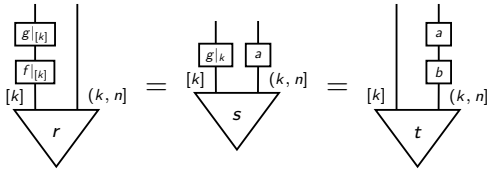
Composability

Theorem

Protocols secure against an attack model $\mathcal{A}$ are closed under composition ($\circ$ and $\otimes$).

Proof.

$\otimes$ and $\circ$ inherited form the ambient category—one just needs to check that they work. Here's the key steps for $\circ$ and $\otimes$ in the n-partite case with the first k parties honest



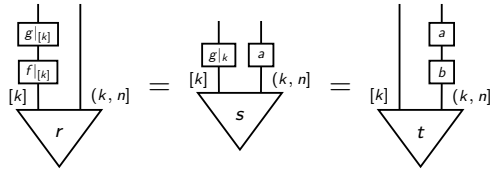
Composability

Theorem

Protocols secure against an attack model $\mathcal{A}$ are closed under composition ($\circ$ and $\otimes$).

Proof.

$\otimes$ and $\circ$ inherited form the ambient category—one just needs to check that they work. Here's the key steps for $\circ$ and $\otimes$ in the n-partite case with the first k parties honest



□

Security against multiple attack models

Corollary

Protocols secure against $\mathcal{A}_1, \ldots \mathcal{A}_k$ form a symmetric monoidal category

Security against multiple attack models

Corollary

Protocols secure against $\mathcal{A}_1, \ldots \mathcal{A}_k$ form a symmetric monoidal category

Proof.

Symmetric monoidal subcategories are closed under intersection □

Security against multiple attack models

Corollary

Protocols secure against $\mathcal{A}_1, \ldots \mathcal{A}_k$ form a symmetric monoidal category

Proof.

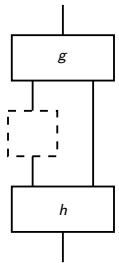
Symmetric monoidal subcategories are closed under intersection □

Example

Fix a family of subsets of $[n]$ parties: protocols secure against each of these subsets behaving maliciously form an SMC. For instance, in MPC one often studies protocols secure against at most $n/2$ or $n/3$ malicious participants.

Resource theory of maps

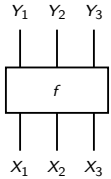
We can easily vary the construction to have our resources be arbitrary morphisms $f : A \to B$ (or $f : \bigotimes_{i=1}^{n} A_i \to \bigotimes_{i=1}^{n} B_i$ in the n-partite case) and our resource conversions be given by (n-tuples of) "combs"



built out of free processes.

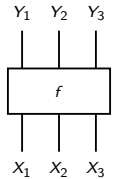
Resource theory of tripartite maps

When $n = 3$, a resource consists of objects $X_1, X_2, X_3, Y_1, Y_2, Y_3$ and of a morphism $f : X_1 \otimes X_2 \otimes X_3 \to Y_1 \otimes Y_2 \otimes Y_3$ in $\mathbf{C}$:

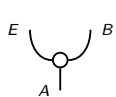


Resource theory of tripartite maps

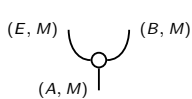
When $n = 3$, a resource consists of objects $X_1, X_2, X_3, Y_1, Y_2, Y_3$ and of a morphism $f : X_1 \otimes X_2 \otimes X_3 \to Y_1 \otimes Y_2 \otimes Y_3$ in $\mathbf{C}$:



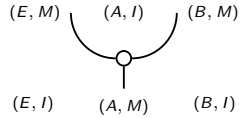
Notational convention: label the three parties as E, A, B (for Eve, Alice and Bob), and label each wire just with its owner. Example:



instead of

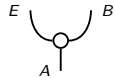


or



OTP: starting resources

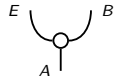
Channel from Alice to Bob that leaks everything to Eve:



(Note: if instead the message goes via Eve (who may tamper with it), the analysis is different)

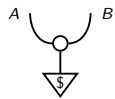
OTP: starting resources

Channel from Alice to Bob that leaks everything to Eve:



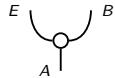
(Note: if instead the message goes via Eve (who may tamper with it), the analysis is different)

Shared random key:



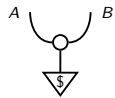
OTP: starting resources

Channel from Alice to Bob that leaks everything to Eve:



(Note: if instead the message goes via Eve (who may tamper with it), the analysis is different)

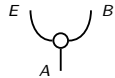
Shared random key:



Target resource: a channel

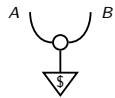
OTP: starting resources

Channel from Alice to Bob that leaks everything to Eve:



(Note: if instead the message goes via Eve (who may tamper with it), the analysis is different)

Shared random key:

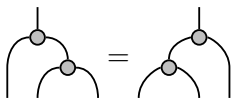


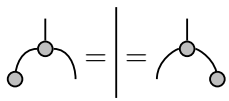
Target resource: a channel

Free building blocks: local (efficient) computation

Local ingredients for OTP

A group structure on the message space: a multiplication  with unit  satisfying the following equations.





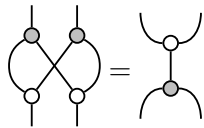
Local ingredients for OTP

A group structure on the message space: a multiplication  with unit  satisfying the following equations.

Note that copying and deleting satisfy similar equations

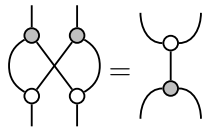
Rest of the group structure

In addition, multiplication and copying interact:

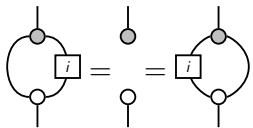


Rest of the group structure

In addition, multiplication and copying interact:

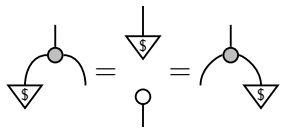


and the map  giving inverses satisfies



Uniform randomness

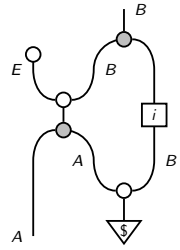
The key being uniformly random is captured by



"Adding uniform noise to a channel gives uniform noise"

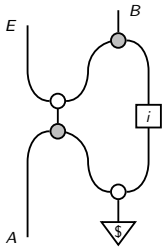
For the experts: a Hopf algebra with an integral in a symmetric monoidal category.

The protocol

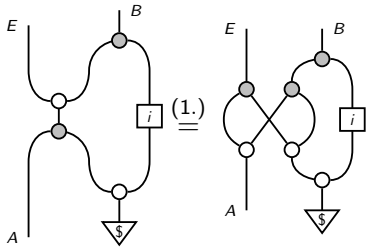


Alice adds the key to her message, broadcasts it to Eve and Bob. Eve deletes her part and Bob adds the inverse of the key to recover the message.

Security of OTP

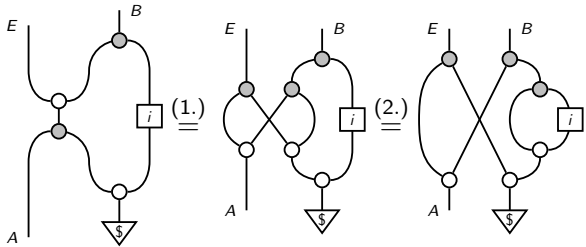


Security of OTP



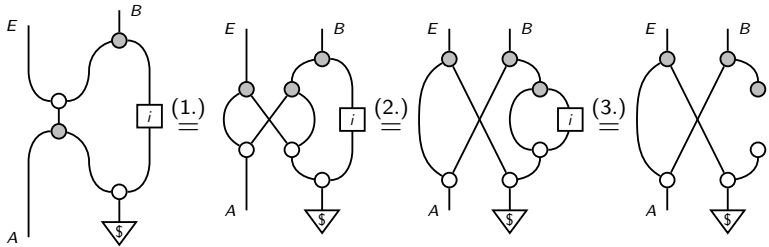
1. Bialgebra.

Security of OTP



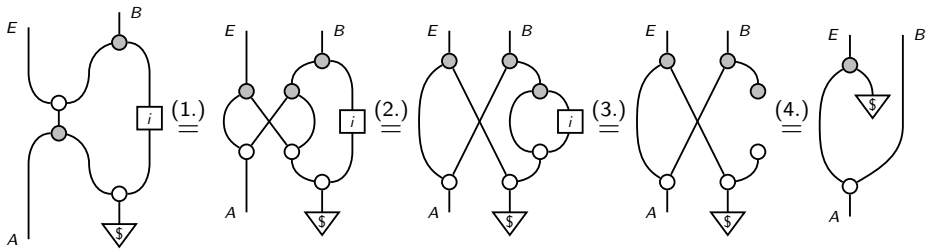
1. Bialgebra. 2. Associativity.

Security of OTP



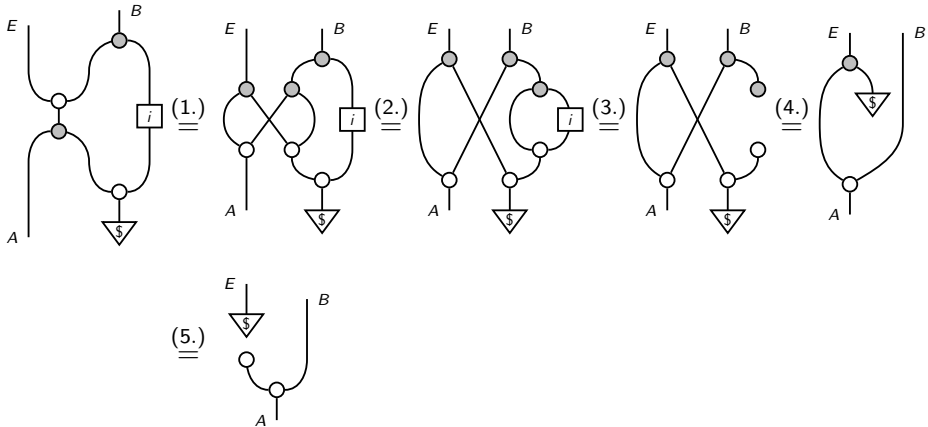
1. Bialgebra. 2. Associativity. 3. Antipode

Security of OTP



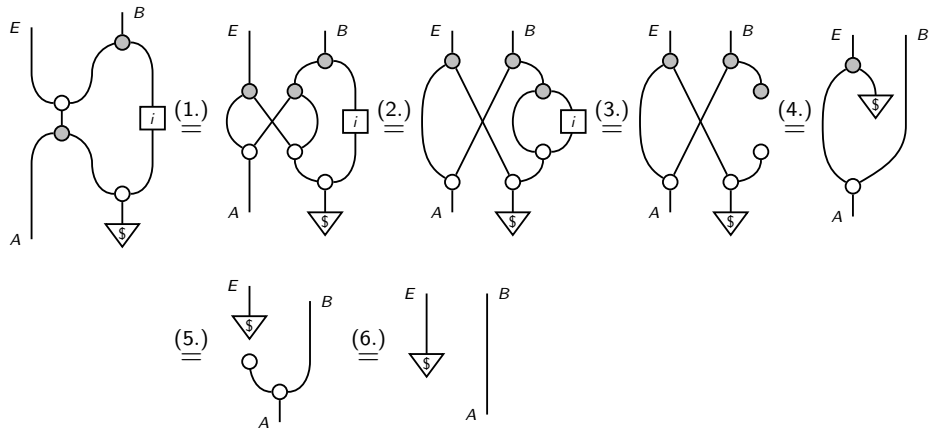
1. Bialgebra. 2. Associativity. 3. Antipode 4. Units

Security of OTP



1. Bialgebra. 2. Associativity. 3. Antipode 4. Units 5. Random noise

Security of OTP



1. Bialgebra. 2. Associativity. 3. Antipode 4. Units 5. Random noise 6. Units.

More on OTP

In other words, anything Eve might learn from the ciphertext she could already compute without it, so this protocol is indeed a secure transformation against Eve.

More on OTP

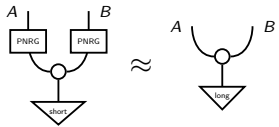
In other words, anything Eve might learn from the ciphertext she could already compute without it, so this protocol is indeed a secure transformation against Eve.

Reusing keys is not a secure map $key \to key \otimes key$.

More on OTP

In other words, anything Eve might learn from the ciphertext she could already compute without it, so this protocol is indeed a secure transformation against Eve.

Reusing keys is not a secure map $key \to key \otimes key$. However, a computationally secure PRNG will give a computationally secure way of constructing a long shared key from a short one, depicted by

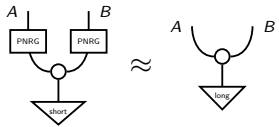


where $\approx$ stands for computational indistinguishability.

More on OTP

In other words, anything Eve might learn from the ciphertext she could already compute without it, so this protocol is indeed a secure transformation against Eve.

Reusing keys is not a secure map $key \to key \otimes key$. However, a computationally secure PRNG will give a computationally secure way of constructing a long shared key from a short one, depicted by



where $\approx$ stands for computational indistinguishability.
Composing these two results in *the stream cipher*, which is secure automatically as a composite of secure protocols inside our framework.

Extensions of the simple model

The above captures a very particular cryptographic situation:
There is no set-up, i.e., the parties have no free cryptographic primitives or communication not given by the starting functionality.

Extensions of the simple model

The above captures a very particular cryptographic situation:
There is no set-up, i.e., the parties have no free cryptographic primitives or communication not given by the starting functionality.

- This can be fixed by fixing a class $\mathcal{X}$ of free resources and defining general protocols $r \to s$ as those of the form $r \otimes x \to s$ with $x \in \mathcal{X}$.

Extensions of the simple model

The above captures a very particular cryptographic situation:
There is no set-up, i.e., the parties have no free cryptographic primitives or communication not given by the starting functionality.

- This can be fixed by fixing a class $\mathcal{X}$ of free resources and defining general protocols $r \to s$ as those of the form $r \otimes x \to s$ with $x \in \mathcal{X}$.

Security is perfect (i.e. information theoretic) instead of computational.

Extensions of the simple model

The above captures a very particular cryptographic situation:
There is no set-up, i.e., the parties have no free cryptographic primitives or communication not given by the starting functionality.

- This can be fixed by fixing a class $\mathcal{X}$ of free resources and defining general protocols $r \to s$ as those of the form $r \otimes x \to s$ with $x \in \mathcal{X}$.

Security is perfect (i.e. information theoretic) instead of computational. This can be fixed in two ways:

- replace $=$ with an equivalence relation $\approx$ modelling computational indistinguishability
- Work with a pseudometric, and work with approximately or asymptotically secure protocols

Further results

Can pass from FinStoch to efficient sequences of stochastic maps, and model Diffie-Hellman key exchange there.

Further results

Can pass from FinStoch to efficient sequences of stochastic maps, and model Diffie-Hellman key exchange there.

Abstract no-go results for bipartite and tripartite security

Further results

Can pass from FinStoch to efficient sequences of stochastic maps, and model Diffie-Hellman key exchange there.

Abstract no-go results for bipartite and tripartite security

Abstract lifting results: strong monoidal functors preserve (some) security

Part 2: Other frameworks in CCC

Pooya Farshim[1 2], Andre Knispel[1], Martti Karvonen[3], Markulf Kohlweiss[1 4] and Phil Wadler[1 4]

[1]IOG

[2]Durham University

[3]University College London

[4]University of Edinburgh

Overview

Goal: formally capture other approaches to composability, such as Universal Composability (UC) or abstract cryptography. Use this to transfer results and ideas between frameworks.

Overview

Goal: formally capture other approaches to composability, such as Universal Composability (UC) or abstract cryptography. Use this to transfer results and ideas between frameworks.

Today: a *sketch* of a base category used to discuss UC.

UC in CCC

An interactive Turing machine (ITMs) has an *identity* and comes with a *communication set* giving the identities of machines it expects input from and sends its outputs to.

UC in CCC

An interactive Turing machine (ITMs) has an *identity* and comes with a *communication set* giving the identities of machines it expects input from and sends its outputs to.

Consider a set P of ITMs with distinct identities and an identity s.

- Assume s is the identity of a machine in P. If it expects input from a machine not in P, then s is an *external subroutine identity*. Otherwise s is an *internal identity*.
- If s is to be sent output from a machine in P but s is not in P, then s is an *external main identity*.

UC in CCC

An interactive Turing machine (ITMs) has an *identity* and comes with a *communication set* giving the identities of machines it expects input from and sends its outputs to.

Consider a set P of ITMs with distinct identities and an identity s.

- Assume s is the identity of a machine in P. If it expects input from a machine not in P, then s is an *external subroutine identity*. Otherwise s is an *internal identity*.
- If s is to be sent output from a machine in P but s is not in P, then s is an *external main identity*.

An open protocol $\approx$ a finite set of ITMs up to renaming internal identities.

UC in CCC

An interactive Turing machine (ITMs) has an *identity* and comes with a *communication set* giving the identities of machines it expects input from and sends its outputs to.

Consider a set P of ITMs with distinct identities and an identity s.

- Assume s is the identity of a machine in P. If it expects input from a machine not in P, then s is an *external subroutine identity*. Otherwise s is an *internal identity*.
- If s is to be sent output from a machine in P but s is not in P, then s is an *external main identity*.

An open protocol $\approx$ a finite set of ITMs up to renaming internal identities.
Given finite sets S and T of identities, a morphism $S \to T$ consists of an open protocol whose external subroutine identities are in S and external main identities are in T.

UC in CCC

An interactive Turing machine (ITMs) has an *identity* and comes with a *communication set* giving the identities of machines it expects input from and sends its outputs to.

Consider a set P of ITMs with distinct identities and an identity s.

- Assume s is the identity of a machine in P. If it expects input from a machine not in P, then s is an *external subroutine identity*. Otherwise s is an *internal identity*.
- If s is to be sent output from a machine in P but s is not in P, then s is an *external main identity*.

An open protocol $\approx$ a finite set of ITMs up to renaming internal identities.
Given finite sets S and T of identities, a morphism $S \to T$ consists of an open protocol whose external subroutine identities are in S and external main identities are in T. Composition by renaming and union. Disjoint union of sets gives rise to a symmetric monoidal structure.

UC in CCC

Some further changes needed:

- Pass to the Kleisli category of the graded monad $\mathbf{C} \to [\mathbf{C}, \mathbf{C}]$ given by $A \mapsto A \otimes -$. An object in this category is a pair of objects of $\mathbf{C}$. A morphism $(X_1, X_2 \to (Y_1, Y_2)$ in this category consists of (an equivalence class of) an object A, and morphisms $X_1 \to A \otimes Y_1$ and $A \otimes X_2 \to Y_2)$. Idea: the object A and the second morphism belongs to the adversary.

UC in CCC

Some further changes needed:

- Pass to the Kleisli category of the graded monad $\mathbf{C} \to [\mathbf{C}, \mathbf{C}]$ given by $A \mapsto A \otimes -$. An object in this category is a pair of objects of $\mathbf{C}$. A morphism $(X_1, X_2 \to (Y_1, Y_2)$ in this category consists of (an equivalence class of) an object A, and morphisms $X_1 \to A \otimes Y_1$ and $A \otimes X_2 \to Y_2$). Idea: the object A and the second morphism belongs to the adversary.

- An n-partite version

UC in CCC

Some further changes needed:

- Pass to the Kleisli category of the graded monad $\mathbf{C} \to [\mathbf{C}, \mathbf{C}]$ given by $A \mapsto A \otimes -$. An object in this category is a pair of objects of $\mathbf{C}$. A morphism $(X_1, X_2 \to (Y_1, Y_2)$ in this category consists of (an equivalence class of) an object A, and morphisms $X_1 \to A \otimes Y_1$ and $A \otimes X_2 \to Y_2$). Idea: the object A and the second morphism belongs to the adversary.

- An n-partite version

- Restrict to a subcategory of machines that behave in a particular way wrt. adversary.

Part 3: Pictures for game-based cryptography

Martti Karvonen[1], Robin Piedeleu[1], Mike Rosulek[2] and Fabio Zanasi[1]

[1]University College London

[2]Oregon State University

Overview

Goal: Understand game-based (not necessarily-composable) cryptography categorically. Use string diagrams for this as well.

Overview

Goal: Understand game-based (not necessarily-composable) cryptography categorically. Use string diagrams for this as well.

Today: A picture proof for the 3-round Feistel, which promotes a pseudorandom function into a pseudorandom permutation.

Some more generators

A (stateful) map 

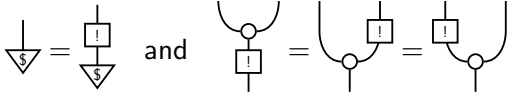
checking that the input is new. Also for pairs 

Some more generators

A (stateful) map 

checking that the input is new. Also for pairs 

Two relevant properties:

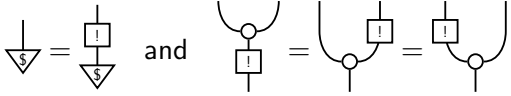


Some more generators

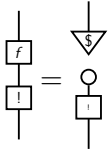
A (stateful) map 

checking that the input is new. Also for pairs 

Two relevant properties:



A pseudorandom function (PRF) $\approx$ a function indistinguishable from a function chosen uniformly at random. Relevant axiom:

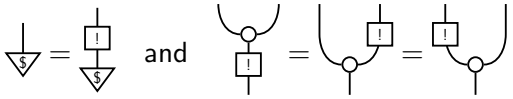


Some more generators

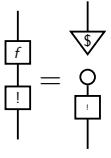
A (stateful) map 

checking that the input is new. Also for pairs 

Two relevant properties:

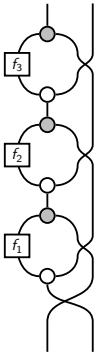


A pseudorandom function (PRF) $\approx$ a function indistinguishable from a function chosen uniformly at random. Relevant axiom:

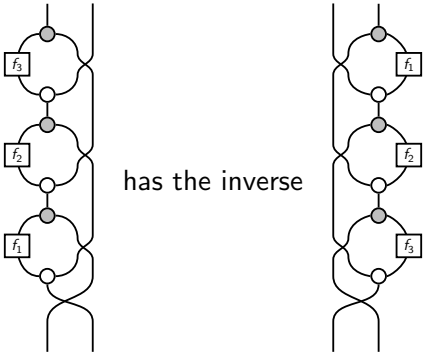


A pseudorandom permutation (PRP) is a PRF that is a permutation (has an inverse)

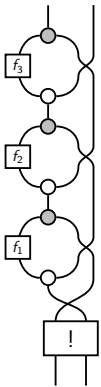
3-round Feistel



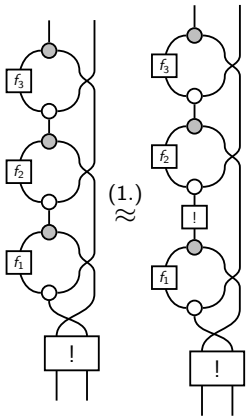
3-round Feistel



3-round Feistel

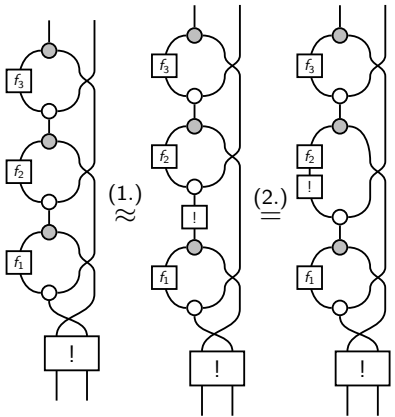


3-round Feistel



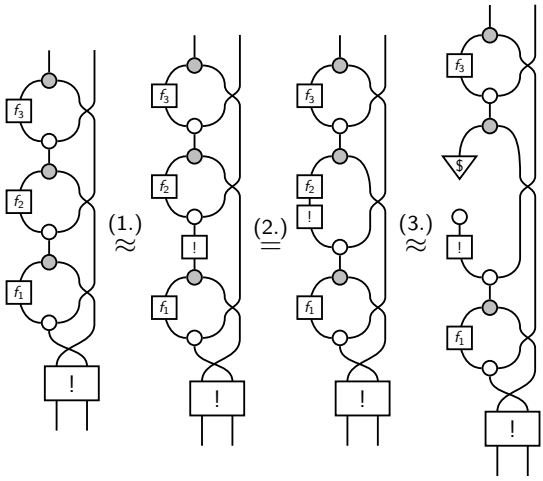
1. Lemma.

3-round Feistel



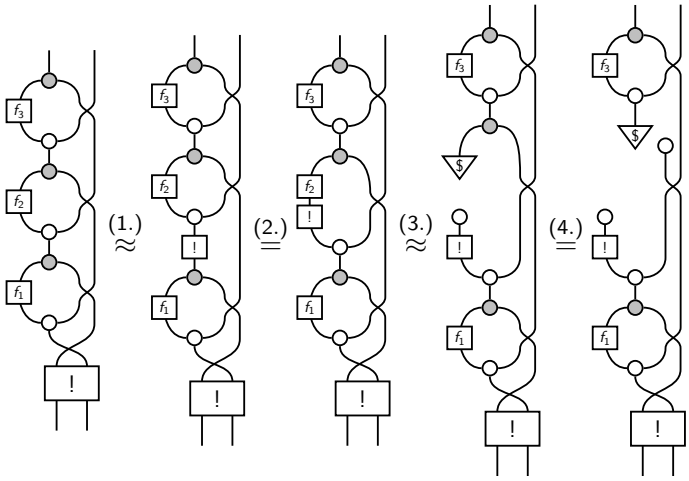
1. Lemma. 2. Sliding !

3-round Feistel



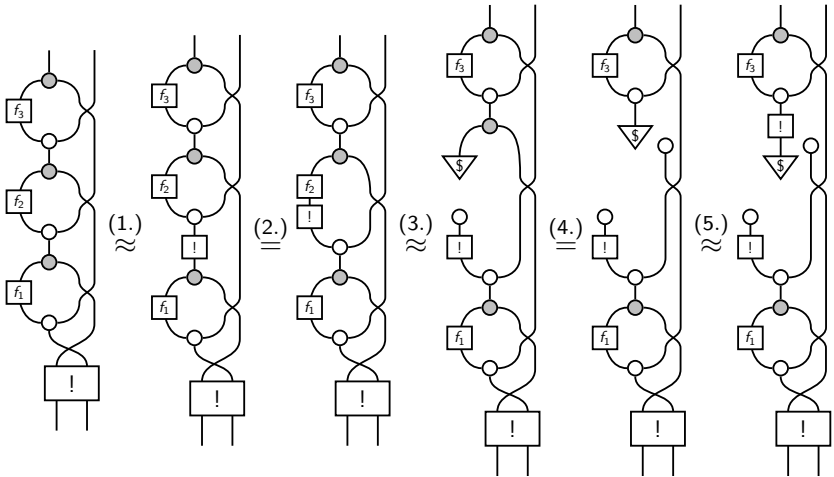
1. Lemma. 2. Sliding ! 3. PRF

3-round Feistel



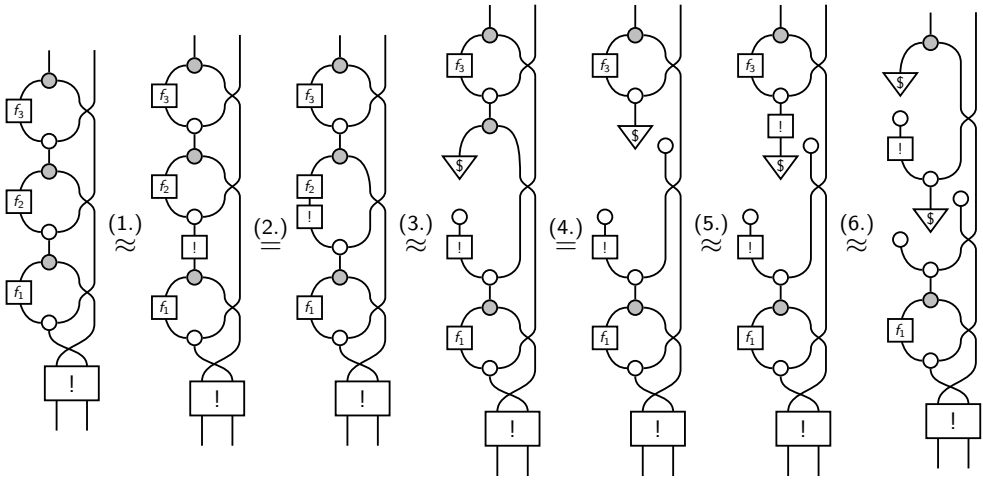
1. Lemma. 2. Sliding ! 3. PRF 4. Randomness

3-round Feistel



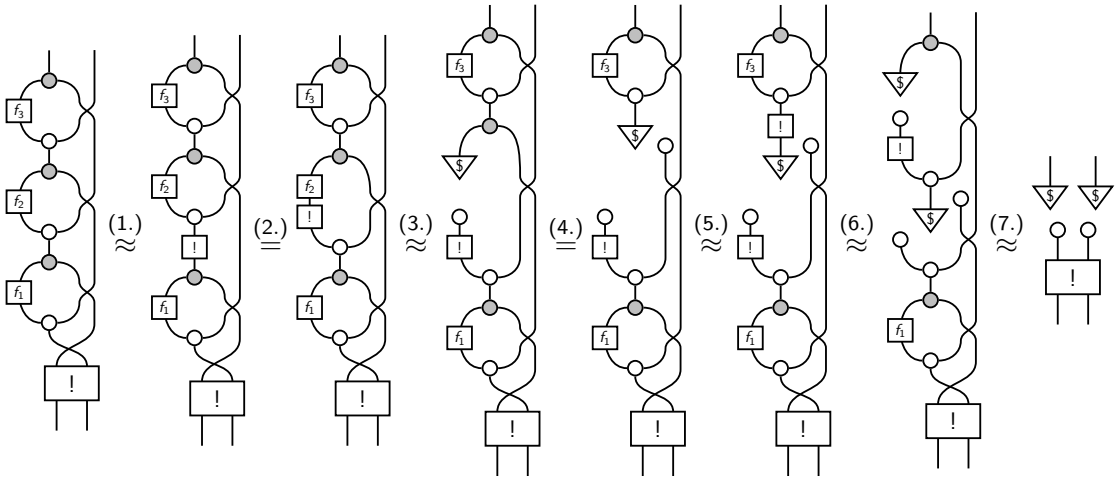
1. Lemma. 2. Sliding ! 3. PRF 4. Randomness 5. B-Day

3-round Feistel



1. Lemma. 2. Sliding ! 3. PRF 4. Randomness 5. B-Day 6. PRF

3-round Feistel



1. Lemma. 2. Sliding ! 3. PRF 4. Randomness 5. B-Day 6. PRF 7. Randomness

Summary

We have a framework where

- composability is guaranteed (also for computational security)
- attack models are general enough to cover various kinds of adversarial behavior (e.g. colluding vs independent attackers)
- string diagrams can be used to make existing (or new) pictures into rigorous proofs

and we're using it (and related ideas) to

- capture other frameworks
- study game-based security

Questions...

?

Broadbent A., MK, “Categorical composable cryptography”, FoSSaCS (2022), arXiv:2105.05949
Broadbent A., MK, “Categorical composable cryptography: extended version”, LMCS (2023), arXiv:2208.13232