Categorical composable cryptography

Martti Karvonen (joint work with Anne Broadbent)

University of Ottawa

INL
21.4.2022

Composability in cryptography

One would expect that if you wire together “provably secure” protocols you end up with a secure protocol.

- This is false in general! Standard game-based security notions don’t necessarily guarantee composability. In fact, many “secure” protocols might not be secure anymore if several copies are run concurrently.
- QKD and 20(ish) years between first security proofs and composable ones.
- Several frameworks for composability and plenty of work within them, but none have convinced the whole community.

Real-world ideal-world paradigm

AKA simulation paradigm.

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 indistuingishable 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 in spirit but technically different

N+1th approach

In our 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
- CT and the tools and connections it brings: in particular, string diagrams

Cryptography as a resource theory

The key idea is to view 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.

We will discuss an extended example - the OTP. Our approach is inspired by

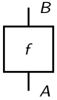
‘*Constructive Cryptography – A New Paradigm for Security Definitions and Proofs*’
Maurer, U., TOSCA 2011.

‘*Bicategorical Semantics for Nondeterministic Computation*’
Stay, M. & Vicary, J., MFPS 2013.

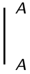
In this viewpoint, OTP is a protocol: $\textit{key} \otimes \textit{insecure channel} \to \textit{secure channel}$

Categories via pictures

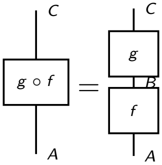
A category has objects and morphisms between them. We will depict a morphism $f\colon A \to B$ by



Special morphisms get special pictures. For plain categories there's only the identity morphisms $\mathrm{id}_A\colon A \to A$ drawn as

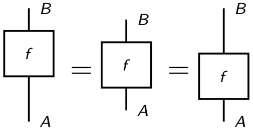


Morphisms can be composed if the object in the middle matches:



Categories via pictures

The pictures make the axioms implicit, or alternatively - the axioms are whatever is needed for the pictures to be unambiguous. Besides associativity, we have left and right unit laws:

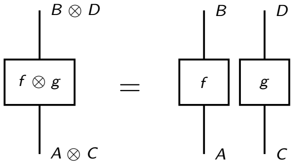


Example

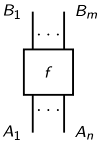
- Sets-with structure and structure-preserving-maps: e.g. sets and functions, topological spaces and cont. maps, vector spaces and linear maps
- Any group gives a category with a single object: morphisms correspond to elements of G and composition is given by multiplication in G
- Any poset $(X, \leq)$ gives a category: elements of X are the objects, and there's a single morphism $x \to y$ iff $x \leq y$

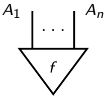
Monoidal categories via pictures

In a *monoidal* category one also has parallel composition of objects and morphism



The monoidal unit is denoted by the empty picture, and boxes can now have arbitrary amounts of input and output wires (a box with no inputs is called a state):

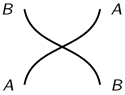




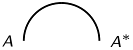
Examples: $(\mathbf{Set}, \times, 1)$, $(\mathbf{Vect}_{\mathbb{F}}, \otimes, \mathbb{F})$

Symmetry and compact closure via pictures

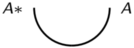
In a *symmetric* monoidal category (SMC) one can also cross wires, and only connectivity matters.



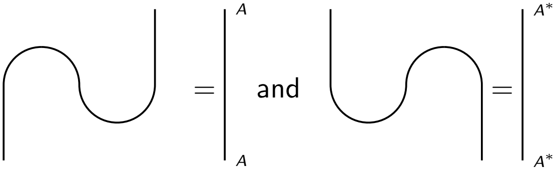
In a compact closed category we also have cups and caps, i.e. morphisms



and



satisfying the snake equations



This blurs the distinction between inputs and outputs. Eg. $(\mathbf{Rel}, \times, 1)$, $(\mathbf{FVect}_{\mathbb{F}}, \otimes, \mathbb{F})$

Resource theories

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?
- Can these (possibly non-local) correlations be used to simulate those?
- Any preordered commutative monoid.

For a category theorist, this is roughly speaking an SMC where you mostly care whether a transformation $A \to B$ exists or not.

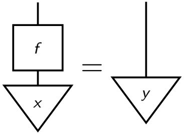
Resource theory of states

In ‘*A mathematical theory of resources*’

Coecke, Fritz & Spekkens, Information and Computation (2016).

many resource theories are built theories starting from an SMC **C** equipped with a (wide) sub-SMC $\mathbf{C}_F$ of free processes.

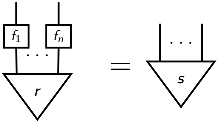
One of the constructions – the resource theory of states – can be desribed as follows: the resources are states of **C** and the resource conversions $x \to y$ are maps f in $\mathbf{C}_F$ such that



.

Resource theory of n-partite states

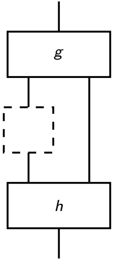
n-partite version: Objects are of the form $((A_i)_{i=1}^n, r\colon I \to \bigotimes A_i)$. A map $((A_i)_{i=1}^n, r) \to ((B_i)_{i=1}^n, s)$ is then a tuple $(f_i)_{i=1}^n$ that transforms r to s:



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

Resource theory of maps

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



built out of free processes.

In general, it suffices to have a "well-behaved"[1] operation R assigining to each object A the set $R(A)$ of resources of type A and to each $f\colon A \to B$ a function $R(f)\colon R(A) \to R(B)$ explaining how f transforms resources.

[1] lax monoidal functor

Security in the abstract

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

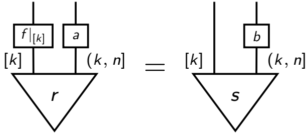
A general resource theory lets us only talk about correctness of transformations. To add in security:

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

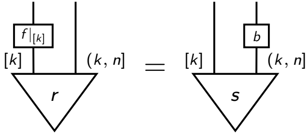
The general theory is developed for an resource theory with an attack model $\mathcal{A}$ satisfying some conditions: today we'll keep things simpler

Dishonest parties

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 $\bigotimes_{k+1}^{n} \mathrm{id}$:



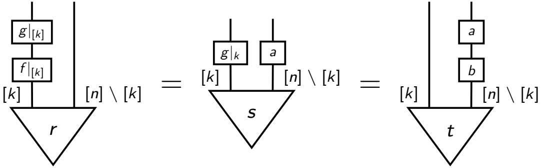
Composability

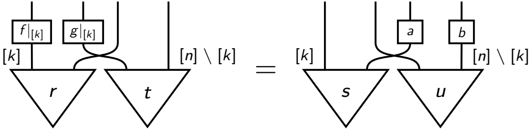
Theorem

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

Proof.

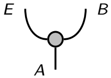
Symmetric monoidal subcategories are closed under intersection □

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.

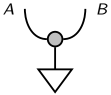
OTP: starting resources

Channel from Alice to Bob that leaks everything to Eve:

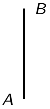


(Note: if instead the message goes via Eve (who may tamper with it), the analysis is different)

Shared random key:



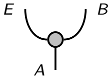
Target resource: a channel



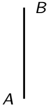
Free building blocks: local (efficient) computation

Insecure protocol

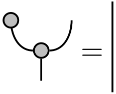
One way of transforming



to



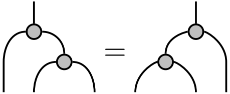
is by having Eve delete everything she receives, as

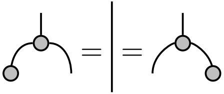


But this is not secure against Eve! No guarantees if Eve disobeys the protocol.

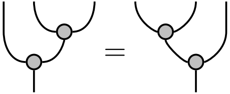
Local ingredients for OTP

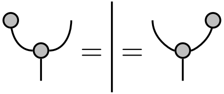
A group structure on the message space: a multiplication with unit satisfying the following equations.





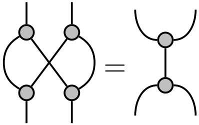
Note that copying and deleting satisfy similar equations



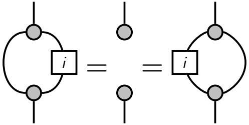


Rest of the group structure

In addition, multiplication and copying interact:

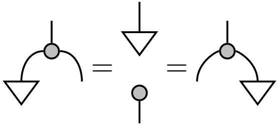


and the map $\boxed{i}$ giving inverses satisfies



Uniform randomness

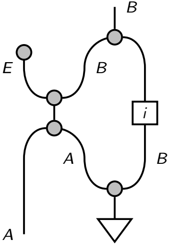
The key being uniformly random is captured by



“Adding uniform noise to a channel gives uniform noise”

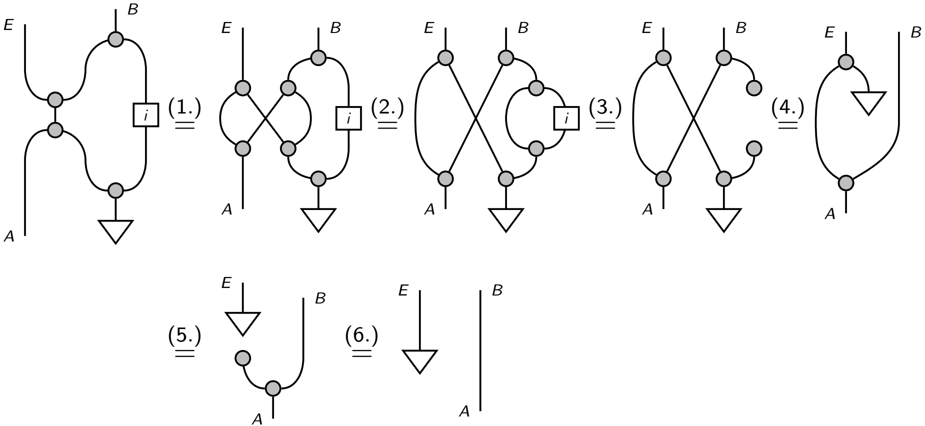
For the experts: a Hopf algebra with an integral in a symmetric monoidal category.

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

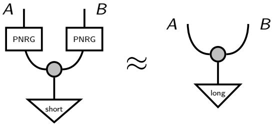


1. Bialgebra. 2. Associativity. 3. Antipode 4. Units 5. Random noise 6. Units.

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.

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 ≈ 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.

- 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 $r \otimes x \to s$.

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

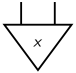
- replace $=$ with an equivalence relation $\approx$ modelling computational indistinguishability
- Work with a pseudometric, and work with approximately or asymptotically secure protocols

A no-go theorem for two parties

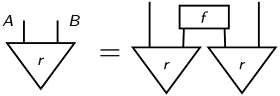
Let **C** now be a compact closed category, with $\cup$ modelling a shared communication channel.

Theorem

For Alice and Bob (one of whom might cheat), if a bipartite functionality r



can be realized from a communication channel between them, i.e. from $\cup$ by a secure protocol, then r satisfies

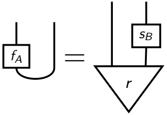


for some f.

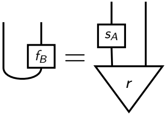
A no-go theorem for two parties

Proof.

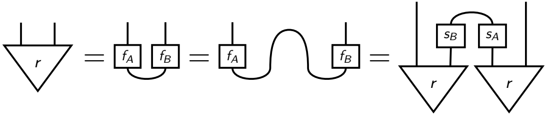
Assume a protocol $f_A \otimes f_B$ achieving this. Security constraints against each party give us



and



Which gives

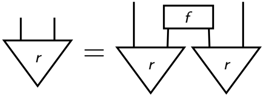


□

A no-go theorem for two parties

Theorem

For Alice and Bob (one of whom might cheat), if a bipartite functionality r can be realized from a communication channel between them, i.e. from $\cup$ *by a secure protocol, then r satisfies*



for some f.

Corollary

In the same bipartite setting, (composable) bit commitment and oblivious transfer are impossible without setup.

Summary

We have a categorical 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

Questions...

?

Broadbent A., MK, “Categorical composable cryptography”, FoSSaCS (2022)

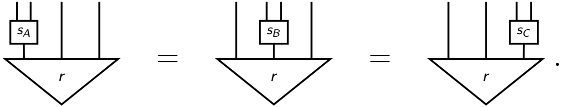
See also:

- my talk at Structure meets power workshop: slides, talk
- my talk at the Applied Category Theory conference: slides, talk

Bonus: tripartite no-go

Theorem

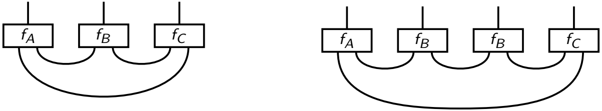
If a tripartite functionality r *can be realized from each pair of parties sharing a state* $\cup$, *securely against any single party, then there are simulators* s_A, s_B, s_C *such that*



Bonus: tripartite no-go

Proof.

Any tripartite protocol building on top of each pair of parties sharing $\cup$ can be drawn as in the left side of

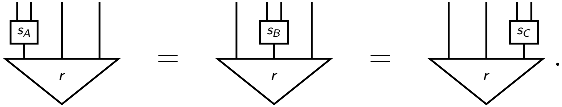


Consider now the picture depicted on the right. It can be seen as the result of three different attacks on the protocol: one where Alice cheats and performs f_A and f_B (and the wire connecting them), one where Bob performs f_B twice, and one where Charlie performs f_B and f_C. The security of (f_A, f_B, f_C) against each of these gives the required simulators. □

Bonus: tripartite no-go

Theorem

If a tripartite functionality r can be realized from each pair of parties sharing a state$\cup$*, securely against any single party, then there are simulators s_A, s_B, s_C such that*



Corollary

Given a SMC **C** *modeling interactive computation, and a state*$\cup$ *on $A \otimes A$ modeling pairwise communication, it is impossible to build broadcasting channels securely (even asymptotically in terms of distinguisher advantage) from pairwise channels.*