RILEY SHAHAR

# TITLE

# *Contents*

*List of Tables*

# *List of Figures*

# 1 Cryptography

## *Resources, Protocols, Adversaries, Attacks*

Cryptography is the mathematical study of secure computation. In a computation, we want to use *protocols* to transform *resources*. For the computation to be secure, it must successfully resist *attacks* by *adversaries*. We will make all of these notions precise, but first we discuss some motivating examples. Many more examples can be found in any introductory text on cryptography, such as [KL14; Ros21; Rap10].

*Notation.* We first need some notation for these examples.

- We write $x \leftarrow\!\!\$\ X$ to mean $x$ is drawn from the distribution $X$; when $X$ is a set we affix the uniform distribution.
- We write $\oplus$ for the bitwise XOR (addition modulo 2) of two binary strings.

**Example 1.1** (The One-Time Pad)**.** Two parties, Alice and Bob, have the same *private key* $k \leftarrow\!\!\$\ \{0,1\}^\ell$ and can communicate over an *insecure channel* $C$. Alice has a *message* $m \in \{0,1\}^\ell$ that she wants to send to Bob. These define the resources required for the following protocol:

One-Time Pad

1 : Alice insecurely sends $c := m \oplus k$ to Bob.

2 : Bob decodes the message as $c \oplus k$.

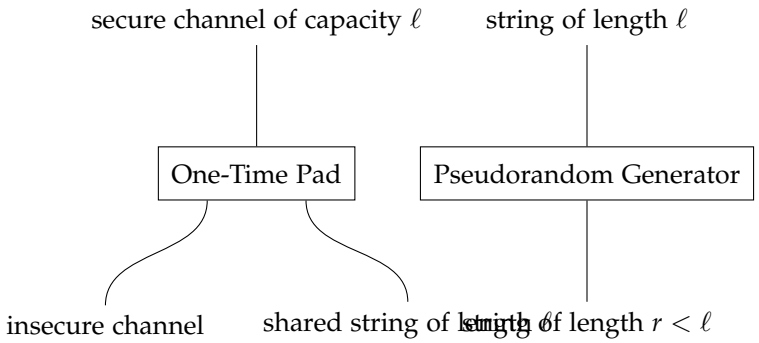
A proof of *correctness* of this protocol should say, roughly, that Bob recieves the same message $m$ that Alice sent.

Since the channel is insecure, a third party, Eve, may observe the ciphertext $c$ sent by Alice. Eve is an *adversary* against the protocol. A proof of *security* of the protocol should say, roughly, that Eve cannot learn anything about $m$ from observing $c$.

In other words, one-time pad is a protocol with two open input ports and one open output port. When "hooked up" to an insecure channel and a shared string of length $\ell$, it produces a secure channel of capacity $\ell$. We can notate this as in **??**.

There are of course many details which we do not communicate in this diagram, for instance that the key must be uniform random.

**Example 1.2** (Pseudorandom Generators)**.** A *pseudorandom generator* should take some small input string, its *seed*, and produce a longer output string, in such a way the its outputs cannot be distinguished from uniform random strings. Again, we will make this formal soon, but for now we express this in **??**.




(a) The One-Time Pad.

(b) A pseudorandom generator.

Figure 1.1: Basic cryptographic protocols.

A limitation of one-time pad is that it needs as long a key as the channel it produces. We should now be able to fix this by *composing* a pseudorandom generator with the one-time pad, as in Figure 1.2.

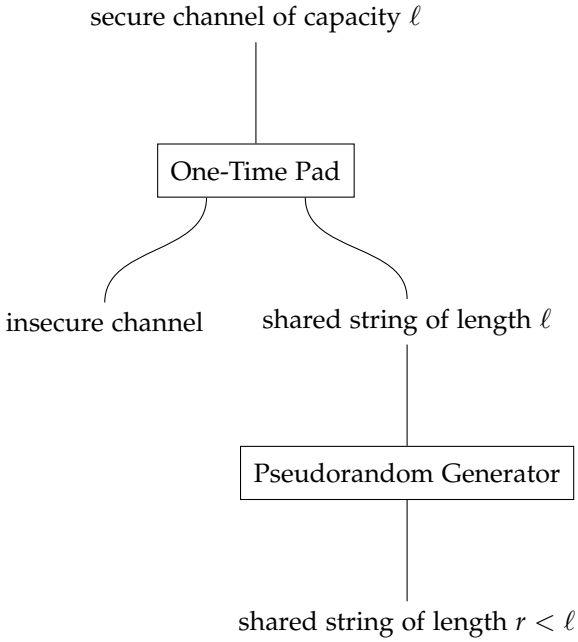



Figure 1.2: Composing One-Time Pad with a pseudorandom generator.

This scheme raises an immediate issue. we need to know that our notions of correctness and security *compose*: if one-time pad and the pseudorandom generator are both correct or both secure, is the composite procotol correct or secure? Notice that, in the composite

diagram, Alice and Bob must both separately use the pseudorandom generator to convert their seed into a longer key. We thus need composition both *in sequence*—"vertically" in the diagram—and *in parallel*—"horizontally" in the diagram.

## *Formulating Cryptographic Protocols*

Cryptographic protocols are procedures for transforming resources. To do math with them, we need a precise definition of what resources they require and produce. For instance, one-time pad is a *private-key encryption scheme*.

**Definition 1.3.** A *private-key encryption protocol* over a *message space* $\mathcal{M}$ and *key space* $\mathcal{K}$ consists of the following data:

- a probabalistic *key generation algorithm* $\mathtt{Gen} : 1^* \to \mathcal{K}$;
- a family of *encryption algorithms* $\mathtt{Enc}_k : \mathcal{M} \to \mathcal{M}$ indexed by $\mathcal{K}$;
- a family of *decryption algorithms* $\mathtt{Dec}_k : \mathcal{M} \to \mathcal{M}$ indexed by $\mathcal{K}$.

By "algorithm", we could have several different notions of computation in mind. Usually we don't require this level of specificity, but it suffices to consider Turing machines.

This definition allows us to straightforwardly formulate the correctness of such a scheme:

**Definition 1.4.** A private-key encryption protocol is *correct* if for all $k \in \mathcal{K}$ and $m \in \mathcal{M}$,

$$\mathtt{Dec}_k(\mathtt{Enc}_k(m)) = m.$$

To say that an encryption protocol is *secure*, we would like to formalize the notion that the adversary learns nothing from intercepting the ciphertext.

## *Game-Based Security*

In *game-based* approaches to security, we define security by determining the winner of an abstract game.

## *Simulation-Based Security*

In *simulation-based* approaches to security, we define security by comparing a protocol in the real world to an ideal world in which the desired resource is produced by a trusted black-box.

# 2 Category Theory

## Categories

**Definition 2.1** (Category)**.** A *category* $\mathcal{C}$ consists of the following data:

- a collection of objects, overloadingly also called $\mathcal{C}$;
- for each pair of objects $x, y \in \mathcal{C}$, a collection of *morphisms* $\mathcal{C}(x, y)$;
- for each object $x \in \mathcal{C}$, a designated *identity morphism* $x \xrightarrow{1_x} x$;
- for each pair of morphisms $x \xrightarrow{f} y \xrightarrow{g} z$, a designated *composite morphism* $x \xrightarrow{gf} z$.

We use the word *collection* for foundational reasons: in many important examples, the objects and morphisms do not form sets. We ignore such foundational issues here; they are discussed in [Mac71, Section 1.6].

This data must satisfy the following axioms:

- *unitality*: for any $x \xrightarrow{f} y$, $1_y f = f = f 1_x$;
- *associativity*: for any $x \xrightarrow{f} y \xrightarrow{g} z \xrightarrow{h} w$, $(hg)f = h(gf)$.

Categories are widespread in mathematics, as the following examples show.

**Example 2.2** (Concrete Categories)**.** The following are all categories:

- Set is the category of sets and functions.
- Grp is the category of groups and group homomorphisms.
- Ring is the category of rings and ring homomorphisms.
- Top is the category of topological spaces and homeomorphisms.
- For any field $\Bbbk$, $\mathrm{Vect}_\Bbbk$ is the category of vector spaces over $\Bbbk$ and linear transformations.

We call such categories, whose objects are structured sets and whose morphisms are structure-preserving set-functions, *concrete*. On the other hand, many categories look quite different.

**Example 2.3.** The following are also categories:

- The *empty category* has no objects and no morphisms.
- The *trivial category* has a single object and its identity morphism.
- Any group (or, more generally, monoid) can be thought of as a category with a single object, a morphism for every element, and composition given by the monoid multiplication.

- Any poset (or, more generally, preorder) $(P, \leq)$ can be thought of as a category whose objects are the elements of $P$, with a unique morphism $x \to y$ if and only if $x \leq y$. In this sense, composition is a "higher-dimensional" transitivity, and identities are higher-dimensional reflexivity.
- Associated to any directed graph is the *free category* on the graph, whose objects are nodes and whose morphisms are paths.
- There is a category whose objects are (roughly) multisets of molecules and whose morphisms are chemical reactions. See [BP17] for a formalization of this notion.

When working with categories, we often want to show that two complex composites equate. In this case, we prefer graphical notation to the more traditional symbolic equalities of Definition 2.1. The key idea is that such diagrams can be "pasted", allowing us to build up complex equalities from simpler ones.

**Definition 2.4** (Commutative Diagram)**.** A diagram *commutes* if, for any pair of paths through the diagram with the same start and end, the composite morphisms are equal.

**Example 2.5.** In this language, the axioms of Definition 2.1 are expressed by commutativity of the diagrams in Figure 2.1.

The notion of a diagram can be made precise fairly easily; see [Rie17, Section 1.6].

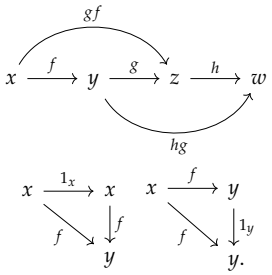



Figure 2.1: The category axioms.

# *Bibliography*

[BK22] 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.

[BP17] John C. Baez and Blake S. Pollard. "A compositional framework for reaction networks". In: *Reviews in Mathematical Physics* 29.09 (Sept. 2017), p. 1750028. DOI: 10.1142/s0129055x17500283. URL: https://doi.org/10.1142%2Fs0129055x17500283.

[CFS16] Bob Coecke, Tobias Fritz, and Robert W. Spekkens. "A mathematical theory of resources". In: *Information and Computation* 250 (2016). Quantum Physics and Logic, pp. 59–86. ISSN: 0890-5401. DOI: https://doi.org/10.1016/j.ic.2016.02.008. URL: https://www.sciencedirect.com/science/article/pii/S0890540116000353.

[GK96] 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.

[KL14] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography, Second Edition*. Chapman & Hall/CRC Cryptography and Network Security Series. Taylor & Francis, 2014. ISBN: 9781466570269. URL: https://books.google.com/books?id=OWZYBQAAQBAJ.

[Lin17] Yehuda Lindell. "How to Simulate It—A Tutorial on the Simulation Proof Technique". In: *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*. Springer International Publishing, 2017, pp. 277–346. ISBN: 9783319570488. DOI: 10.1007/978-3-319-57048-8_6.

[Mac71] Saunders MacLane. *Categories for the Working Mathematician*. Graduate Texts in Mathematics, Vol. 5. New York: Springer-Verlag, 1971, pp. ix+262.

[Rap10] "Abhi Shelat" "Raphael Pass". *A Course in Cryptography*. 2010. URL: https://www.cs.cornell.edu/courses/cs4830/2010fa/lecnotes.pdf.

[Rie17] Emily Riehl. *Category theory in context*. en. Courier Dover Publications, Mar. 2017.

[Ros21] Mike Rosulek. *The Joy of Cryptography*. 2021. URL: https://joyofcryptography.com.