Title

A Thesis
Presented to
The Established Interdisciplinary Committee
for Mathematics and Computer Science
Mathematics and Natural Sciences
Reed College

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Arts

Riley Shahar

May 200x

Approved for the Division
(Mathematics)

Angélica Osorno

Adam Groce

Acknowledgements

Preface

List of Abbreviations

Table of Contents

List of Tables

List of Figures

Abstract

Dedication

Introduction

Chapter 1

Cryptography

1.1 Resources, Protocols, Adversaries

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, RP10].

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

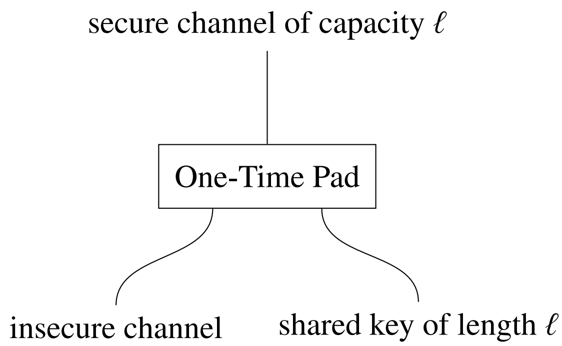
Alice insecurely sends $c := m \oplus k$ to Bob.

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.

Given an insecure channel and a shared private key, this protocol defines a single-use, secure channel from Alice to Bob. We can notate this as follows:



In other words, One-Time Pad is a protocol with two open input ports and one open output port[1]. When "hooked up" to an insecure channel and a shared key of length ℓ, it produces a secure channel of capacity ℓ.

Example 1.2 (Pseudorandom Generators)**.** A *pseudorandom generator*

1.2 Simulation-Based Security

If cryptograph is the mathematical study of secure computation, a first question is precisely what we mean by 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.

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

Chapter 2

Category Theory

2.1 Categories

Definition 2.1 (Category)**.** A *category* C consists of the following data:

- a collection[1] of objects, overloadingly also called C;
- for each pair of objects $x, y \in C$, a collection of *morphisms* $C(x, y)$;
- for each object $x \in C$, 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* $x \xrightarrow{gf} z$.

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:

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

[1] 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].

- 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) groups 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.

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

Example 2.5. In this language, the axioms of Definition 2.1 are expressed by commutativity of the following diagrams:

$$
\begin{tikzcd}
x \arrow[r, "f"] \arrow[rr, bend left, "gf"] & y \arrow[r, "g"] \arrow[rr, bend right, "hg"'] & z \arrow[r, "h"] & w
\end{tikzcd}
\qquad
\begin{tikzcd}
x \arrow[r, "1_x"] \arrow[dr, "f"'] & x \arrow[d, "f"] \\
& y
\end{tikzcd}
\qquad
\begin{tikzcd}
x \arrow[r, "f"] \arrow[dr, "f"'] & y \arrow[d, "1_y"] \\
& y.
\end{tikzcd}
$$

References

[BK22] Anne Broadbent and Martti Karvonen. Categorical composable cryptography. In *Foundations of software science and computation structures*, volume 13242 of *Lecture Notes in Comput. Sci.*, pages 161–183. Springer, Cham, 2022.

[BP17] John C. Baez and Blake S. Pollard. A compositional framework for reaction networks. *Reviews in Mathematical Physics*, 29(09):1750028, sep 2017.

[KL14] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography, Second Edition*. Chapman & Hall/CRC Cryptography and Network Security Series. Taylor & Francis, 2014.

[Lin17] Yehuda Lindell. *How to Simulate It—A Tutorial on the Simulation Proof Technique*, pages 277–346. Springer International Publishing, 2017.

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

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

[Ros21] Mike Rosulek. *The Joy of Cryptography*. 2021.

[RP10] ”Abhi Shelat” ”Raphael Pass”. *A Course in Cryptography*. 2010.