RILEY SHAHAR

TITLE

Contents

1 *Cryptography* 7

2 *Category Theory* 11

List of Tables

List of Figures

2.1 The category axioms. 11

1 Cryptography

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 a motivating example.

Many more examples can be found in any introductory text on cryptography, such as [KL14; Ros21; Rap10].

Commitment Protocols

Suppose we want to build an online rock-paper-scissors game. If two players, Alice and Bob, want to play, then they should both be able to send each other a move and so determine the winner. However, something needs to prevent the players from waiting until after they learn the other's move to choose their own move. This is an ideal use-case for a *commitment protocol*.

Informally, a commitment protocol proceeds as follows. The sender S has a message m that they wish to commit to—in our case, m is one of $\{R, P, S\}$. In the *commit phase*, they send a commitment c to the receiver R. At some later time, S may reveal m—plus maybe some auxillary data, for example their random bits—to R, in which case R should be able to verify that c was indeed a commitment to m.

We can formalize commitment schemes as follows:

Definition 1.1 (commitment protocol)**.** A *commitment protocol* consists of the following data:

- the *message space* $\mathcal{M}$ and *commit space* $\mathcal{C}$;
- a pair of families of probabalistic interactive algorithms S_n and R_n (indexed by the *security parameter* $n \in \mathbb{N}$), respectively called the *sender* and *receiver*, such that:
  - in the *commit phase*, S_n gets $m \in \mathcal{M}$ and returns $c \in \mathcal{C}$, while R_n returns $\bot$ or $\top$;
  - in the *reveal phase*, S_n gets $m \in \mathcal{M}$, while R_n gets $c \in \mathcal{C}$, returning $\bot$ or $m' \in \mathcal{M}$.

We have been imprecise about our formal notion of algorithm. For our purposes in this chapter, Turing machines suffice; we will be more precise about this later. In particular, it will be important that we consider the security parameter as indexing a family of algorithms, rather than as a unary input to each algorithm as is common in the literature.

Notation. Let S and R be interactive algorithms as in Definition 1.1.

- We will write $\mathtt{Com}_S^R(m)$ for the output of S in the commit phase with S getting input $m \in \mathcal{M}$, or $\perp$ if R returns $\perp$.
- We will write $\mathtt{Rev}_S^R(m; c)$ for the output of R in the reveal phase with S getting input $m \in \mathcal{M}$ and R getting input $c \in \mathcal{C}$.

  Where S and R are clear, we may omit the respective annotations.

We would like the commitment scheme to be correct, in that when the parties behave honestly according to the protocol, the receiver returns the correct message. Formally:

Definition 1.2. A commitment protocol (S_n, R_n) is *correct* if for all $n \in \mathbb{N}$ and $m \in \mathcal{M}$,

$$\Pr[\mathtt{Rev}_{S_n}^{R_n}(m; \mathtt{Com}_{S_n}^{R_n}(m)) = m] = 1.$$

When n is unbound, we use this notation to indicate a family of pairs $\{(S_n, R_n) : n \in \mathbb{N}\}$; when n is bound it refers to the specific pair (S_n, R_n).

Given a commitment protocol (S_n, R_n), we should be able to define a family of algorithms for our rock-paper-scissors game as follows. Let $\mathcal{M} = \{R, P, S\}$ and let $W : (\perp \sqcup \mathcal{M})^2 \to \{1, 0, -1, \perp\}$ compute whether the first argument beats, ties, or loses to the second, propagating any $\perp$s.

Protocol 1.3 (Rock-Paper-Scissors).

1. A_n receives input $a \in \mathcal{M}$; B_n receives input $b \in \mathcal{M}$.
2. A_n acts as S_n and B_n as R_n to compute $c_a = \mathtt{Com}_{S_n}^{R_n}(a)$.
3. A_n acts as R_n and B_n as S_n to compute $c_b = \mathtt{Com}_{S_n}^{R_n}(b)$.
4. A_n acts as S_n and B_n as R_n to compute $a' = \mathtt{Rev}_{S_n}^{R_n}(a; c_a)$.
5. A_n acts as R_n and B_n as S_n to compute $b' = \mathtt{Rev}_{S_n}^{R_n}(b; c_b)$.
6. A_n returns $W(a, b')$.
7. B_n returns $W(a', b)$.

Notation. Given some fixed commitment protocol, we will write $\mathtt{RPS}_A^B(a, b)$ for the results returned by A and B, respectively. If A or B are honest, we will omit the corresponding annotation.

As with commitment, we can define correctness of this protocol.

Definition 1.4. Protocol 1.3 is *correct* relative to a commitment protocol (S_n, R_n) if for all $a, b \in \{R, P, S\}$,

$$\Pr[\mathtt{RPS}(a, b) = (W(a, b), W(a, b))] = 1.$$

Theorem 1.5. *Protocol 1.3 is correct relative to any correct commitment protocol.*

Proof. Fix a correct commitment protocol. Then

$$\begin{aligned}
&\Pr[\mathtt{RPS}(a,b) = (W(a,b), W(a,b))] \\
&= \Pr[W(a,b') = W(a,b) \text{ and } W(a',b) = W(a,b)] \\
&= \Pr[W(a,b') = W(a,b)] \cdot \Pr[W(a',b) = W(a,b)] \\
&\geq \Pr[b' = b] \cdot \Pr[a' = a] \\
&= \Pr[\mathtt{Rev}(b; c_b) = b] \cdot \Pr[\mathtt{Rev}(a; c_a) = a] \\
&= \Pr[\mathtt{Rev}(b; \mathtt{Com}(b)) = b] \cdot \Pr[\mathtt{Rev}(a; \mathtt{Com}(a)) = a] \\
&= 1.
\end{aligned}$$

□

At least in this case, it was easy to define notions of correctness that compose. Our task now is to define security of commitment and rock-paper-scissors such that, whenever a commitment scheme is secure, Protocol 1.3 is likewise secure.

Game-Based Security

In *game-based* approaches to security, we define security by determining the winner of an abstract game. Here, we encode specific properties we want the algorithm to have, and say that an adversary wins the game if they can break the property. In standard approaches to commitment, there are two desirable properties. *Hiding* means that the receiver should not learn anything about the message until the reveal phase. *Binding* means that the sender should not be able to trick the receiver into anything other than the committed message. Formally:

See [Cano8; Golo1, Section 4.4.1].

Definition 1.6. A commitment scheme (S_n, R_n) is *game-secure* if the following hold.

- *Hiding:* consider the following game against a family of adversaies R'_n.

  > *Game* 1.7.
  >
  > 1. R'_n outputs $m_0, m_1 \in \mathcal{M}$.
  > 2. A random bit $b \in \{0,1\}$ is chosen; m_b is given to S_n.
  > 3. S_n and R'_n participate in the commitment phase.
  > 4. R'_n outputs a guess $b' \in \{0,1\}$. R'_n wins if $b' = b$.

  A commitment scheme is *hiding* if for any family of adversaries R'_n,

  $$\Pr[R'_n \text{ wins Game 1.7}] \leq \frac{1}{2} + \mathsf{negl}(n).$$

  The idea is that R' participates in the commitment phase for a randomly chosen message m_b, and then tries to guess b; they should be able to guess no more than negligibly better than random.

A function is *negligible*, written $\mathsf{negl}(n)$, if it is asymptotically smaller than any rational function in n.

- *Binding:* consider the following game against a family of adversaires S'_n.

  Game 1.8.

  1. S'_n outputs $m_0, m_1 \in \mathcal{M}$.
  2. S'_n and R_n participate in the commitment phase.
  3. A random bit $b \in \{0,1\}$ is chosen and given to S'_n.
  4. S'_n and R_n participate in the reveal phase.
  5. S'_n wins if R_n outputs m_b.

  A commitment scheme is *binding* if for any family of adversaries S'_n,

$$\Pr[S'_n \text{ wins Game 1.8}] \leq \frac{1}{2} + \mathsf{negl}(n).$$

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* $x \xrightarrow{gf} z$.

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.

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].

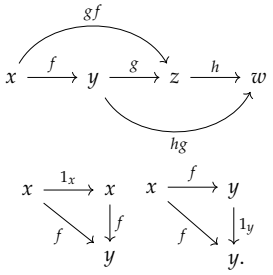

Figure 2.1: The category axioms.

- 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.

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 diagrams in Figure 2.1.

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.

[Can08] Ran Canetti. *Lecture 11*. Spring 2008. URL: https://www.cs.tau.ac.il/~canetti/f08-materials/scribe11.pdf.

[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.

[Gol01] O. Goldreich. *Foundations of Cryptography: Volume 1, Basic Tools*. Cambridge University Press, 2001. ISBN: 9780521791724.

[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.

[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.