Chasing diagrams in cryptography

To Jim Lambek for his 90th birthday

Dusko Pavlovic[*]
Email: `dusko.pavlovic@rhul.ac.uk`

Royal Holloway, University of London

Abstract. Cryptography is a theory of secret functions. Category theory is a general theory of functions. Cryptography has reached a stage where its structures often take several pages to define, and its formulas sometimes run from page to page. Category theory has some complicated definitions as well, but one of its specialties is taming the flood of structure. Cryptography seems to be in need of high level methods, whereas category theory always needs concrete applications. So why is there no categorical cryptography? One reason may be that the foundations of modern cryptography are built from probabilistic polynomial-time Turing machines, and category theory does not have a good handle on such things. On the other hand, such foundational problems might be the very reason why cryptographic constructions often resemble low level machine programming. I present some preliminary explorations towards categorical cryptography. It turns out that some of the main security concepts are easily characterized through *diagram chasing*, going back to Lambek's seminal *'Lecture Notes on Rings and Modules'*.

1   Introduction

1.1   Idea

For a long time, mathematics was subdivided into geometry and arithmetic, later algebra. The obvious difference between the two was that the geometric reasoning was supported by pictures and diagrams, whereas the algebraic reasoning relied upon the equations and abstract text. For various reasons, the textual reasoning seemed dominant in XX century mathematics: there were relatively few pictures in the mathematical publications, and even the formal systems for geometry were presented as lists of formulas. But as the algebraic constructions grew more complex, the task to stratify and organize them grew into a mathematical problem on its own. Category theory was proposed as a solution for this problem. The earliest categorical diagrams expanded the textual reasoning from exact

---

[*] Recent primary affiliation: University of Hawaii at Manoa. Email: `dusko@hawaii.edu`

sequences to matrices of exact exact sequences [22,7]. The technique of diagram chasing seem to have emerged around the time of Lambek's classic "Lectures on Rings and Modules" [19], where it was used not just as a convenient visualization of lists of equations, but also as a geometric view of universal constructions. This unassuming idea then proceeded to form a germ of geometric reasoning in category theory, uncovering the geometric patterns behind abstract logical structures [23]. Other forms of geometric reasoning emerged in various forms of categorical research [14,12,6, to mention just a few], providing some of the most abstract algebraic structures with some of the most concrete geometric tools.

The present paper reports about the beginnings of an exploration towards applying categorical diagrams in a young and exciting area of mathematics: modern cryptography. Initiated in the late 1970s [3] by introducing algorithmic hardness as a tool of security, modern cryptography developed a rich conceptual and technical apparatus in a relatively short period of time. The increasing complexity of its proofs and constructions, usually presented in a textual, "command line" mode, akin to low-level programming, occasionally engendered doubts that its formalisms may sometimes conceal as many errors as they prevent [16,15,17]. Would a high level categorical view help?

1.2 Background

Modern cryptography is a theory of effectively computable, randomized boolean functions. A boolean function is a mapping over bitstrings, i.e. in the form $f : 2^M \longrightarrow 2^N$, where $2 = \{0, 1\}$ denotes the set of two elements, and M, N are finite sets. So 2^M denotes the set of M-tuples of 0 and 1; or equivalently of the subsets of M. Which view of 2^M is more convenietn depends on the application. Formally, the algebraic structure of 2^M is induced by the algebraic structure of 2, which is usually viewed as

- Boolean algebra $(2, \wedge, \vee, \neg, 0, 1)$
- Boolean ring $(\mathbb{Z}_2, \oplus, \cdot, 0, 1)$
- submonoid $\{1, -1\} \subseteq (\mathbb{Z}_3, \cdot)$

A boolean function f is *effectively computable*, or *feasible*, and denoted by $f : 2^M \xrightarrow{\mathcal{F}} 2^N$, when it is implemented by a boolean circuit, a Turing machine with suitable time and space bounds, or in some other model of computation. Computations in general are, of course, generally expressed as effective boolean functions over the representations of mathematical structures by bitstrings, all the way up to the continuum [27].

A *randomized* boolean function $g : 2^M \xrightarrow{\mathcal{R}} 2^N$ is in fact a boolean function of two arguments, say $g : 2^R \times 2^M \longrightarrow 2^N$, where the first argument is interpreted as a *random seed*. The output of a randomized function is viewed as a random variable. The probability that a randomized boolean function g, given an input

x produces an output y is estimated by counting for how many values of the random seed ρ it takes that value, i.e.

$$\Pr(y \leftarrow gx) = \frac{\#\{\rho \in 2^R \mid y = g(\rho, x)\}}{2^R}$$

where $\#S$ denotes the number of elements of the set S, and R is the length of the random seeds ρ.

An *effective, randomized* boolean function $h : 2^M \xrightarrow{\mathcal{RF}} 2^N$ is thus an effectively computable boolean function $h : 2^R \times 2^M \xrightarrow{\mathcal{F}} 2^N$. It is usually realized by a Deterministic Polynomial-time Turing (DPT) machine, i.e. as $h : 2^R \times 2^M \xrightarrow{DPT} 2^N$. A DPT with two input tapes, one of which is interpreted as providing the random seeds, is called a Probabilistic Polynomial-time Turing (PPT) machine. So for the same function h we would write $h : 2^M \xrightarrow{PPT} 2^N$, leaving the random seeds implicit. This is what cryptographers talk about in their formal proofs, although they seldom specify any actual PPTs. Building a PPT is tedious work, in fact an abstract form of low level machine programming. For a high level view of cryptographic programming, an abstract theory of feasible functions is needed.

Before we proceed in that direction, let us quickly summarize what cryptographers actually build from effective randomized boolean functions and PPTs.

A *crypto system* is a structure given over three finite sets

- $\mathcal{M}$ of *plaintexts*
- $\mathcal{C}$ of *cyphertexts*
- $\mathcal{K}$ of *keys*

plus a set of random seeds, that we leave implicit. They are all given with their bitstring representations. The structure of the crypto-system consists of three feasible functions

- key generation $\langle k, \overline{k} \rangle : 1 \xrightarrow{PPT} \mathcal{K} \times \mathcal{K}$,
- encryption $\mathsf{E} : \mathcal{K} \times \mathcal{M} \xrightarrow{PPT} \mathcal{C}$, and
- decryption $\mathsf{D} : \mathcal{K} \times \mathcal{C} \xrightarrow{DPT} \mathcal{M}$,

that together provide

- unique decryption: $\mathsf{D}(\overline{k}, \mathsf{E}(r, k, m)) = m$,
- and secrecy.

This secrecy is in fact what cryptography is all about. Even defining it took a while.

The earliest formal definition of secrecy is due to Shannon [29]. His idea was to require that the ciphertext discloses nothing about the plaintext. He viewed the attacker as a statistician, equipped with the precise frequency distribution of the language of the meaningful expressions in $\mathcal{M}$, i.e. knowing exactly the values of $\Pr(m \leftarrow \mathcal{M})$, the probability that a randomly sampled string from $\mathcal{M}$ is the plaintext m. Shannon's requirement was that knowing the encryption $c = \mathsf{E}(r, k, m)$ should not make it any easier for this attacker to guess m, or formally

$$\Pr\left(m \leftarrow \mathcal{M} \mid \exists rk.\ c = \mathsf{E}(r, k, m)\right) = \Pr\left(m \leftarrow \mathcal{M}\right) \tag{1}$$

Shannon wrote this in a different, but equivalent form[1], and called it *perfect security*.

When the age of modern cryptography broke out, the concept of secrecy got refined by considering the feasibility of the encryption and decryption operations, and moreover strengthened by requiring that the attacker is unlikely to guess not only the plaintext m, but even a single bit from it. Otherwise, the concept of secrecy would miss the possibility that the plaintext is hard to guess as a whole, but that it may be easy to guess bit by bit. The original formalization of this requirement is due to Goldwasser and Micali [9,10] under the name *semantic security*, but it was later somewhat simplified to the form of *chosen plaintext indistinguishability* (IND-CPA), which looks something like this:

$$\Pr\left(b \leftarrow \mathsf{A}_1(m_0, m_1, c, s) \mid c \leftarrow \mathsf{E}(k, m_b),\ b \leftarrow 2,\ m_0, m_1, s \leftarrow \mathsf{A}_0\right) \sim \frac{1}{2} \tag{2}$$

The attacker consists of two PPTs, A_0 and A_1, which communicate through a tape. He tests the crypto system as follows. First A_0 chooses and announces two plaintexts m_0 and m_1. She may also convey to A_1 a part of her state, by writing s on their shared tape. Then the crypto system tosses a fair coin b, computes the encryption $c \leftarrow \mathsf{E}(k, m_b)$ of one of the chosen plaintexts, and gives it to the attacker. The attacker A_1 is now supposed to guess which of the two plaintexts was encrypted. The system is secure if knowing c does not give him any advantage in this, i.e. if his chance to guess b is indistinguishable from $\Pr(b \leftarrow 2) = \frac{1}{2}$.

The point that I am trying to make is that this is mouthful of a definition. Especially when we are defining secrecy, which is one of the most basic concepts of cryptography. The upshot is that the most basic cryptographic proofs need to show that some crypto system satisfies the above property.

It is, of course, not unheard of that the fundamental concepts tend to be subtle, and require complicated formal definitions. In cryptography, however, this phenomenon seems to be escalating. First of all, the above definition of secrecy as chosen plaintext indistinguishability turns out to be too weak, and too simple. In reality, the attacker can usually access a decryption oracle, which she can consult before she chooses any plaintexts, and also after she receives back the

---

[1] Except that the encryption was not randomized at the time.

encryption of one of them, but before she attempts to guess which one it is. So the attacker actually consists of four PPTs, A_0, A_1, A_2 and A_3, where A_0 begins with choosing some cyphertexts, which it submits to the decryption oracle, etc. A reader who is not a cryptographer may enjoy deciphering the interactions between the crypto system and the attacker from the formula below, describing the *chosen cyphertext indistinguishability* (IND-CCA2), due to Rackoff and Simon [28]. The PPTs again share a tape, which they can use to pass each other a part of the state, denoted s_0, s_1 etc.

$$
\Pr \left(b \leftarrow A_3(c_0, m, m_0, m_1, c, c_1, \widetilde{m}, s_2) \; \middle| \; \begin{array}{l} m = D(\overline{k}, c_0), \; c_0, s_0 \leftarrow A_0, \\ c \leftarrow E(k, m_b), \; b \leftarrow 2, \; m_0, m_1, s_1 \leftarrow A_1(c_0, m, s_0) \\ \widetilde{m} = D(\overline{k}, c_1), \; c_1, s_2 \leftarrow A_2(c_0, m, m_0, m_1, c^{\neq}, s_1) \end{array} \right) \;\; \sim \;\; \frac{1}{2} \quad (3)
$$

This formula is nowadays one of the centerpieces of cryptography. As verbose as it may look, and as prohibitive as its requirements may be[2], it came to be a solid and useful concept. The problem is, however, that the story does not end with it, and that the concepts of ever greater complexity and verbosity rapidly proliferate. This makes cryptographic proofs fragile, with some errors surviving extensive examination [30]. The argument that mandatory formal proofs, if they are too complex, may decrease, rather than increase, the reliability of the proven statements, by decreasing the expert scrutiny over the proven statements, while concealing subtle errors, has been raised from within the cryptographic community [1,2,15,16,17]. At the same time, the efforts towards the formalization have ostensibly consolidated the field and clarified some of its conceptual foundations [8,13]. Maybe we have good reasons and enough insight to start looking for better notations?

Outline of the paper

Section 2 presents a symbolic model of a crypto system, and a very crude symbolic definition of secrecy. These definitions can be stated in any relational calculus, and thus also in the category of relations. Section 3 presents an information theoretic model of a crypto system. The symbolic definition of secrecy refines here to Shannon's familiar definition of perfect security. We formalize it all in the category of sets and stochastic operators between them. And finally, Section 4 introduces a category where the modern cryptographic concepts can be formalized, such as (IND-CPA) and (IND-CCA2). The upshot of this development is to

---

[2] The attacker may submit, e.g. two very large plaintexts, say video blocks, as m_0 and m_1. After she receives the encryption c of one of them, she can then flip just one bit of it, and make that into c_1, which is submitted back for decryption. Although c and c_1 differ in a single bit, the decryption of c_1 should not disclose even a single bit of information about c_0.

show how the incremental approach, refining the crude abstract concepts, while enriching the categorical structures, motivates the conceptual development and provides technical tools. Section 5 invites for further work.

2 Symbolic cryptography

In [5,4], Dolev, Yao, Even and Karp describe public key cryptosystems using an algebraic theory — roughly what mathematicians would call *bicyclic semigroups* [11].

2.1 Dolev-Yao crypto systems

Definition 2.1. *A* message algebra $\mathcal{A}$ *consists of three operations:*

- *encryption* $\mathsf{E} : \mathcal{A} \times \mathcal{A} \longrightarrow \mathcal{A}$,
- *decryption* $\mathsf{D} : \mathcal{A} \times \mathcal{A} \longrightarrow \mathcal{A}$, *and*
- *key pairing* $\overline{(-)} : \mathcal{A} \longrightarrow \mathcal{A}$,

and one equation:

$$\mathsf{D}\left(\overline{k}, \mathsf{E}(k,m)\right) = m$$

called decryption condition*. By convention, the first arguments of* E *and* D *are called* keys*, the second arguments* messages*. A message that occurs in* E *is a* plaintext*; a message that occurs in* D *is a* cyphertext*.*

Definition 2.2. *A* Dolev-Yao crypto system *is given by*

- *a message algebra*
- *a set $M \subseteq \mathcal{A}$ of* well-formed *plaintexts;*
- *the* hiding condition*: "knowing $E(k,m)$ does not reveal anything about m"*

Remarks. The above definitions are close in spirit to Dolev and Yao's definitions, but deviate in details from their presentation. First of all, Dolev and Yao do not present the encryption and decryption operations as binary operations, but as families of unary operations indexed by the keys. More importantly, their results also require the *encryption equation*

$$\mathsf{E}\left(k, \mathsf{D}(\overline{k}, c)\right) = c$$

that should hold for all keys k and all ciphertexts c. Nowadays even toy crypto systems do not satisfy this, so we allow that $\mathsf{E}(k,-)$ may not be surjective. Restricted to its image, of course, the decryption equation implies the encryption

equation; but not generally. Finally, Dolev and Yao do not take $M \subseteq \mathcal{A}$ as a part of the structure. Intuitively, if $\mathcal{A}$ is the set of character strings, then M may be construed as the set of the meaningful words meaningful of some language. For a cryptographer, the capability to distinguish the meaningful words, and recognize a decryption when he finds it, is often critical. The set $M \subseteq \mathcal{A}$ is thus a first, very crude step towards the concepts of source redundancy and frequency distribution, which are of course crucial for cryptanalysis.

The main challenge left behind Dolev and Yao's analysis is that the hiding condition, which is clearly the heart of the matter, is left completely informal. At the first sight, there seem to be many ways to make it precise. We present one in the next section. Its conceptual analogy with the more familiar information theoretic and computational notions of secrecy are clear, but its technical utility seems limited.

2.2 Algebraic perfect security

An attacker sees a ciphertext c and wants to know the plaintext m, such that $\mathsf{E}(k, m) = c$. But since she does not know the key k, she can only form the set of possible[3] plaintexts m that may correspond to c

$$c\widetilde{D} = \{m \in M \mid \exists k.\ \mathsf{E}(k, m) = c\} \qquad (4)$$

One way to formalize the hiding condition is to require that any well-formed message m must be a candidate for a decryption of c, and thus lie in $c\widetilde{D}$.

Definition 2.3. *A Dolev-Yao crypto system $\mathcal{A}$ is algebraically perfectly secure if every ciphertext can be an encryption of any well-formed message, i.e. if for all $c, m \in \mathcal{A}$ holds*

$$m \in M\ \wedge\ \exists k \in \mathcal{A}.\ \mathsf{E}(k, m) = c \iff m \in M \qquad (5)$$

The following lemma says that this captures the intended requirement that the set $c\widetilde{D}$ does not tell anything about m.

Lemma 2.1. *A Dolev-Yao crypto system $\mathcal{A}$ is algebraically perfectly secure if and only if for all $c, m \in \mathcal{A}$ and the binary relation $\widetilde{D}$ from (4) holds*

$$c\widetilde{D}m \iff m \in M \qquad (6)$$

---

[3] I.e., this is the only thing that she can do in the *possibilistic* world of mere relations. In the *probabilistic* world of stochastic relations, she can of course do more, and that will be discussed in the next section.

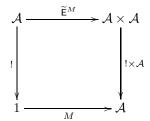A convenient framework to work with algebraic security is the category Rel of sets and binary relations

$$|\mathsf{Rel}| = |\mathsf{Set}|$$
$$\mathsf{Rel}(A, B) = \{0, 1\}^{A \times B}$$

with the usual relational composition of $A \xrightarrow{R} B$ and $B \xrightarrow{S} C$

$$a(R; S)c \iff \exists b \in B.\ aRb \wedge bSc$$

and the equality $aIb \iff a = b$ as the identity relation $A \xrightarrow{I} A$. Note that any subset, say $M \subseteq \mathcal{A}$, can be viewed as a relation $M \in \{0, 1\}^{1 \times \mathcal{A}}$, where $1 = \{0\}$, and thus as an arrow $1 \xrightarrow{M} \mathcal{A}$ in Rel with $0Mx \iff x \in M$.

Proposition 2.4. *A Dolev-Yao crypto system $\mathcal{A}$ is algebraically perfectly secure if and only if the following diagram commutes in the category of relations* Rel

where

- $\mathcal{A} \xrightarrow{!} 1$ *denotes the total relation, i.e. $x!0$ holds for all x and $1 = \{0\}$, and*
- $\widetilde{\mathsf{E}}^M$ *is by definition $c\,\widetilde{\mathsf{E}}^M(k, m) \iff m \in M \wedge \mathsf{E}(k, m) = c$.*

3 Information theoretic cryptography

Shannon [29] brought cryptography to the solid ground of information theory, recognizing the fact that an attacker has access not just to the set $M \subseteq \mathcal{A}$ of possible plaintexts, but also to their probabilities $\mu : \mathcal{A} \longrightarrow [0, 1]$. And just like we viewed the former one in the form $M \in \{0, 1\}^{1 \times \mathcal{A}}$ as an arrow $1 \xrightarrow{M} \mathcal{A}$ in Rel, we shall now view the latter, in the form $\mu \in [0, 1]^{1 \times \mathcal{A}}$ as an arrow $1 \xrightarrow{\mu} \mathcal{A}$ in the category Sto of stochastic matrices.

3.1 Shannon crypto systems

To begin, Shannon introduced into analysis *mixed* crypto systems, in the form $R = pS + (1 - p)T$ where S and T can be thought of as two Dolev-Yao crypto

systems, and $p \in [0,1]$. The idea is that the system R behaves like S with probability p, and like T with probability $1-p$. In summary, Shannon considered message algebras $\mathcal{A}$

(a) given with a probability distribution $\mu : \mathcal{A} \longrightarrow [0,1]$ that assigns to each plaintext m a *frequency*, $\mu(m)$, and moreover
(b) convex closed, in the sense that for any $p \in [0,1]$

$$\mathsf{E}\left(pk + (1-p)h, m\right) = p\mathsf{E}(k,m) + (1-p)\mathsf{E}(h,m)$$
$$\mathsf{D}\left(pk + (1-p)h, m\right) = p\mathsf{D}(k,m) + (1-p)\mathsf{D}(h,m)$$

But (b) makes it convenient to draw the keys from the convex hull of $\mathcal{A}$

$$\Delta\mathcal{A} = \left\{ \kappa : \mathcal{A} \longrightarrow [0,1] \mid \#\varsigma\kappa < \infty \ \wedge \ \sum_{x \in \varsigma\kappa} \kappa(x) = 1 \right\}$$

where $\varsigma\kappa = \{x \in \mathcal{A} \mid \kappa(x) > 0\}$ is the support. As a consequence, the encryption and decryption maps are not functions any more, but stochastic matrices E^κ and D^κ with the entries

$$\mathsf{E}^\kappa_{cm} = \Pr_\kappa(c|m) \ = \ \sum_{\substack{x \in \varsigma\kappa \\ \mathsf{E}(x,m)=c}} \kappa(x)$$

$$\mathsf{D}^\kappa_{mc} = \Pr_\kappa(m|c) \ = \ \sum_{\substack{x \in \varsigma\kappa \\ \mathsf{D}(\overline{x},c)=m}} \kappa(x)$$

Condition (a) similarly suggests that a plaintext, or the available partial information about it, should also be viewed as a stochastic vector $\mu \in \Delta\mathcal{A}$. A crypto system is now an algebra in the category of sets and stochastic operators

$$|\mathsf{Sto}| = |\mathsf{Set}|$$
$$\mathsf{Sto}(M,N) = \left\{ \varPhi \in [0,1]^{M \times N} \mid \#\varsigma\varPhi < \infty \ \wedge \ \sum_{i \in \varsigma\varPhi} \varPhi_{ij} = 1 \right\}$$

Indeed, the encryption and the decryption operations are now stochastic operators $\mathsf{E}^\kappa, \mathsf{D}^\kappa \in \mathsf{Sto}(\mathcal{A}, \mathcal{A})$; whereas the mixed plaintexts are the points $\mu \in \mathsf{Sto}(1, \mathcal{A})$.

Definition 3.1. *A* Shannon crypto system *is given by*

- *a message algebra in the category* Sto*, i.e. stochastic operators for*
  - *encryption* $\mathsf{E} : \mathcal{A} \times \mathcal{A} \longrightarrow \mathcal{A}$*,*
  - *decryption* $\underline{\mathsf{D} : \mathcal{A} \times \mathcal{A}} \longrightarrow \mathcal{A}$*, and*
  - *key pairing* $\overline{(-)} : \mathcal{A} \longrightarrow \mathcal{A}$*,*
- *a frequency distribution of the plaintexts* $\mu : \mathcal{A} \longrightarrow [0,1]$*, and*
- *the hiding condition.*

This time, the formal definition of the hiding condition available, and well known.

3.2 Perfect security

Shannon [29] considers an attacker who makes a *probabilistic* model of the observed crypto system. More precisely, when she observes a cyphtertext c, instead of forming the set $c\widetilde{D} \subseteq M$ of *possible* decryptions, like in Sec. 2.2, she now tries to compute the conditional distribution $\Pr(m|c) \geq \Pr(m)$ of the *probable* decryptions of c.

But now the cyphertext c is a random variable $\gamma = \Pr(c)$, which can be viewed as an arrow $1 \xrightarrow{\Pr(c)} \mathcal{A}$ in Sto. An observation of a cyphertext thus provides knowledge about the distribution of $\Pr(c)$. We assume that the attacker knows the distribution $\kappa = \Pr(k)$ of the keys, and the frequency distribution $\mu = \Pr(m)$.

Definition 3.2. *A Shannon crypto system is perfectly secure if the plaintexts are statistically independent on the cyphertexts, i.e. if for all $c, m \in \mathcal{A}$ holds*
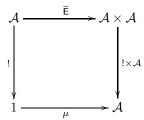
$$\Pr\left(m \leftarrow \mu \mid \exists k.\ c = \mathsf{E}(k, m)\right) = \Pr(m \leftarrow \mu) \tag{7}$$

where the conditional probability on the left stands for

$$\Pr\left(m \leftarrow \mu \mid \exists k.\ c = \mathsf{E}(k, m)\right) = \sum_{x \in \varsigma\kappa} \Pr\left(m \leftarrow \mu \mid c = \mathsf{E}(x, m)\right) \cdot \kappa(x)$$

Aligning definitions 2.3 and 3.2 shows that algebraic perfect security is an algebraic approximation of Shannon's probabilistic perfect security [29, II.10]. The following proposition shows that the connection extends to the categorical characterizations.

Proposition 3.3. *A Shannon crypto system $\mathcal{A}$ with finite support is perfectly secure if and only if the following diagram commutes in the category of stochastic operators Sto*

where

- $\mathcal{A} \xrightarrow{!} 1$ *is the row vector of $\frac{1}{\#\varsigma\mathcal{A}}$,*
- $1 \xrightarrow{\mu} \mathcal{A}$ *is the distribution μ viewed as a column vector,*
- $\mathcal{A} \times \mathcal{A} \xrightarrow{!\times\mathcal{A}} \mathcal{A}$ *is the stochastic matrix with the entries*

$$(!\times\mathcal{A})_{i(jk)} = \begin{cases} \frac{1}{\#\varsigma\mathcal{A}} & \text{if } i = k \\ 0 & \text{otheriwise} \end{cases}$$

10

$-$ $\widetilde{\mathsf{E}}$ *is the stochastic matrix with the entries*

$$\widetilde{\mathsf{E}}_{c(km)} = \begin{cases} \kappa(k) \cdot \mu(m) & \textit{if } c = \mathsf{E}(k, m) \\ 0 & \textit{otherwise} \end{cases}$$

4 Computational cryptography

Modern cryptography arose from the idea to use computational complexity as a tool, and attacker's computational limitations as the persistent assumptions upon which the cryptographer can built the desired security guarantees. To represent modern crypto system, we need to lift the preceding considerations beyond the mere frequency distributions and randomness, captured in the category Sto, to a category suitable to represent randomized feasible computations, graded by a security parameter.

4.1 Category of effective stochastic ensembles up to indistinguishability

The category suitable to present cryptographic constructions will be build by incremental refinement of the category of sets and functions, in three steps: we first make functions feasible, then randomize them, and finally capture the security parameter.

Effective functions Suppose that every set is given with an encoding: e.g., each element is encoded as a bitstring. A function between encoded sets can then be considered feasible if it is realized by a feasible boolean function on the codes.

Let us begin with a crude realization of this idea, just to get a feeling for it. Let $R = (2^*)^{2^*}$ be the monoid of boolean functions and $F \subseteq R$ a submonoid of functions that we call *feasible*. For concreteness, we could assume that the functions from F are just those realized by some suitable family of boolean circuits or Turing-machines. The category Set_F of F-computable functions is then defined

$$|\mathsf{Set}_F| = |\mathsf{Set}/2^*| \;=\; \sum_{A \in |\mathsf{Set}|} \{[\![-]\!]_A : A \longrightarrow 2^*\}$$

$$\mathsf{Set}_F(A, B) = \{f \in \mathsf{Set}(A, B) \mid \exists \varphi \in F \; \forall a \in A. \; [\![f(a)]\!]_B = \varphi[\![a]\!]_A\}$$

$$
\begin{array}{ccc}
A & \xrightarrow{\;f\;} & B \\
{\scriptstyle [\![-]\!]_A} \downarrow & & \downarrow {\scriptstyle [\![-]\!]_B} \\
2^* & \xrightarrow[\;\varphi\;]{} & 2^*
\end{array}
$$

Effective substochastic operators Now we want to refine the category Set_F effective functions to a category of *randomized* effective functions. The step is analogous to the step from Set to Sto. So randomized effective functions will actually be effective stochastic operators. But since feasible functions may not be total, we will actually work with effective *sub*stochastic operators.

The first task is to define the monoid of randomized boolean functions that will operate on the codes. Consider the set of partial functions

$$\mathcal{R} = \big\{\gamma : 2^* \times 2^* \rightharpoonup 2^* \mid \forall x \forall \rho_1 \forall \rho_2.\ \gamma(\rho_1, x){\downarrow}\ \wedge\ \gamma(\rho_2, y){\downarrow}\ \wedge\ |x| = |y|$$
$$\implies\ |\rho_1| = |\rho_2|\ \wedge\ |\gamma(\rho_1, x)| = |\gamma(\rho_2, y)| \big\}$$

where $f(x){\downarrow}$ asserts that the partial function f is defined at x, and $|\xi|$ denotes the length of the bitstring ξ. The set $\mathcal{R}$ forms a monoid $(\mathcal{R}, \circ, \iota)$ where

$$\gamma \circ \beta(\rho_2 :: \rho_1, x) = \gamma(\rho_2, \beta(\rho_1, x)) \tag{8}$$

and $\iota(\langle\rangle, x) = x$, where $\langle\rangle$ denotes the empty string. This monoid was previously used in [26]. Let $\mathcal{F} \subseteq \mathcal{R}$ be a submonoid of functions that we consider feasible. An example are the functions realized by DPT machines. The category $\mathsf{Sto}_\mathcal{F}$ of effective substochastic operators is now defined as follows

$$|\mathsf{Sto}_\mathcal{F}| = |\mathsf{Set}/2^*|\ \ =\ \sum_{A \in |\mathsf{Set}|} \big\{ [\![-]\!]_A : A \longrightarrow 2^* \big\}$$
$$\mathsf{Sto}_\mathcal{F}(A, B) = \big\{ \Phi \in [0,1]^{A \times B} \mid \exists \varphi \in \mathcal{F}\ \forall a \in A\ \forall b \in B.$$
$$\Phi_{ab} = \Pr\left([\![b]\!]_B \leftarrow \varphi[\![a]\!]_A\right) \big\}$$

Ensembles In order to capture security parameters, we must expand randomized functions to ensembles. A *feasible ensemble* is a sequence of feasible functions

$$\psi = \Big\{ \psi_\ell : 2^{r(\ell)} \times 2^{s(\ell)} \xrightarrow{\mathcal{F}} 2^{t(\ell)} \mid \ell \in \omega \Big\}$$

where $\omega = \{0, 1, 2, \ldots\}$, and such that

$$k < \ell \implies \psi_k = \psi_\ell {\upharpoonright}_{(2^{r(k)} \times 2^{s(k)})}\ \ \wedge\ \ r(k) < r(\ell)\ \wedge\ s(k) < s(\ell)\ \wedge\ t(k) < t(\ell)$$

Write $\mathcal{F}^\omega$ for the set of feasible ensembles. A typical example of an ensamble is the extensional (i.e. input-output) view of a PPT machine, which can consume longer inputs, and then it produces longer outputs.

The monoid structure on $\mathcal{F}^\omega$ is induced by the monoid structure of $\mathcal{F}$. The composite $\vartheta \circ \psi$ of

$$\vartheta = \Big\{ \vartheta_k : 2^{u(\ell)} \times 2^{v(\ell)} \xrightarrow{\mathcal{F}} 2^{w(\ell)} \mid \ell \in \omega \Big\}\ \text{ and}$$
$$\psi = \Big\{ \psi_\ell : 2^{r(\ell)} \times 2^{s(\ell)} \xrightarrow{\mathcal{F}} 2^{t(\ell)} \mid \ell \in \omega \Big\}$$

consists of the components

$$(\vartheta \circ \psi)_\ell = \overline{\vartheta}_{\overline{\ell}} \circ \psi_\ell : 2^{\overline{u}(\ell)} \times 2^{\overline{v}(\ell)} \xrightarrow{\mathcal{F}} 2^{t(\ell)}$$

where

- $\overline{\ell}$ is the smallest number such that $w(\overline{\ell}) \geq s(\ell)$,
- $\overline{\vartheta}_{\overline{\ell}} = \vartheta_{\overline{\ell}} \upharpoonright_{2^{s(\ell)}}$,
- $\overline{\vartheta}_{\overline{\ell}} \circ \psi_\ell$ is defined by (8),
- $\overline{u}(\ell) = u(\overline{\ell})$ and $\overline{v}(\ell) = v(\overline{\ell})$.

The category $\mathsf{Sto}^\omega_{\mathcal{F}}$ of effective substochastic ensembles is now defined as follows

$$|\mathsf{Ens}_{\mathcal{F}}| = |\mathsf{Set}/2^\omega| \;\; = \sum_{A \in |\mathsf{Set}|} \{[\![-]\!]_A : A \longrightarrow 2^\omega\}$$

$$\mathsf{Ens}_{\mathcal{F}}(A, B) = \big\{\Psi \in [0,1]^{\omega \times A \times B} \mid \exists \psi \in \mathcal{F}^\omega \; \forall \ell \in \omega \; \forall a \in A \; \forall b \in B.$$
$$\Psi^\ell_{ab} = \Pr\left([\![b]\!] \leftarrow \psi_\ell[\![a]\!]\right)\big\}$$

where

$$\Pr\left([\![b]\!] \leftarrow \psi_\ell([\![a]\!])\right) = \frac{\#\left\{\rho \in 2^{r(\ell)} \mid [\![b]\!]_{t(\ell)} = \psi_\ell\left(\rho, [\![a]\!]_{s(\ell)}\right)\right\}}{2^{r(\ell)}}$$

In the special case when $\mathcal{F}^\omega$ consists of the actions of PPT machines, we get the category $\mathsf{Ens}_{\mathrm{PPT}}$, where the morphisms are the extensional views of PPTs. More precisely, a morphism is a sequence of substochastic matrices $\Psi = \{\Psi^\ell\}_{\ell \in \omega}$ such that there is a PPT Π and the ab-entry of Ψ^ℓ is $\Psi^\ell_{ab} = \Pr(b \leftarrow \Pi_\ell a)$, where ℓ is the security parameter.

So $\mathsf{Ens}_{\mathcal{F}}$ comes close to providing an abstract view of the universe in which the cryptographers work. The view is abstract in the sense that $\mathcal{F}$ does not have to be realized by PPTs, but can be any submonoid of $\mathcal{R}$. By taking $\mathcal{F}$ to be the PPT realized stochastic operations we get the usual probabilistic algorithms — *except* that those that are indistinguishable, because their difference is a negligible function still correspond to different morphisms in $\mathsf{Ens}_{\mathrm{PPT}}$.

Indistinguishability Note, first of all, that $[0,1]$ is not only a monoid, but an ordered semiring[4]. The semiring structure lifts to $[0,1]^\omega$. A *semi-ideal* in an ordered semiring is a lower closed subset closed under addition and multiplication. Since it is lower closed, it contains 0, but generally not 1.

Let $\Upsilon \subseteq [0,1]^\omega$ be a semi-ideal. The canonical example is the semi-ideal of *negligible functions* [8]. A function $\nu : \omega \longrightarrow [0,1]$ is called negligible if $\nu(x) <$

---

[4] A *semiring* is a structure $(R, +, \cdot, 0, 1)$ such that $(R, +, 0)$ and $(R, \cdot, 1)$ are commutative monoids such that $a(b + c) = ab + ac$ and $a0 = 0$.

$\frac{1}{q(x)}$ holds eventually, for every positive polynomial q. Any semi-ideal Υ induces on $[0,1]^\omega$ the equivalence relation

$$\sigma \underset{\Upsilon}{\sim} \tau \iff \exists \nu \in \Upsilon.\ |\sigma_\ell - \tau_\ell| < \nu(\ell)$$

and we define $\mathsf{Ens}_{\mathcal{F}}^{\Upsilon}$ to be the category with the same objects as $\mathsf{Ens}_{\mathcal{F}}$, but

$$\mathsf{Ens}_{\mathcal{F}}^{\Upsilon}(A, B) = \mathsf{Ens}_{\mathcal{F}}(A, B)\big/ \underset{\Upsilon}{\sim}$$

Unfolding this definition over the semring $\mathcal{J}_\Upsilon = [0,1]^\omega / \underset{\Upsilon}{\sim}$, we have

$$\mathsf{Ens}_{\mathcal{F}}^{\Upsilon}(A, B) = \big\{ \Psi \in \mathcal{J}_\Upsilon^{A \times B} \mid \exists \psi \in \mathcal{F}^\omega\ \forall \ell \in \omega\ \forall a \in A\ \forall b \in B.$$
$$\Psi_{ab}^\ell = \mathrm{Pr}\left(\llbracket b \rrbracket \leftarrow \psi_\ell \llbracket a \rrbracket\right) \big\}$$

4.2 Characterizing semantic security

The usual definition of a crypto system from the Introduction can now be stated abstractly, in a categorical form. While the definition follows the pattern of Def. 2.2 and Def. 3.1, this time we revert to the usual *multi-sorted* specification, where the plaintexts, the cyphertexts and the keys are drawn from different sets.

Definition 4.1. *An* abstract crypto system, *relative to a monoid $\mathcal{F}$ of feasible functions, and a semi-ideal Υ of negligible functions is given by*
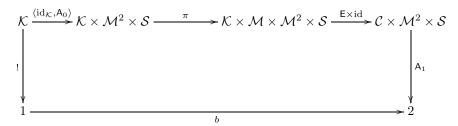
- *a multi-sorted message algebra in the category $\mathsf{Ens}_{\mathcal{F}}^{\Upsilon}$, such that*
  - *encryption $\mathsf{E} : \mathcal{K} \times \mathcal{M} \longrightarrow \mathcal{C}$, is a stochastic ensemble, whereas*
  - *decryption $\underline{\mathsf{D}} : \mathcal{K} \times \mathcal{C} \longrightarrow \mathcal{M}$, and*
  - *key pairing $\overline{(-)} : \mathcal{K} \longrightarrow \mathcal{K}$ are deterministic functions[5].*
- *a frequency distribution of the plaintexts $\mu : \mathcal{M} \longrightarrow [0,1]$, and*
- *the hiding condition.*

The upshot of it all. The abstract versions of the hiding conditions, such as (IND-CPA) and (IND-CCA2), described in the Introduction, boil down to commutative diagrams in $\mathsf{Ens}_{\mathcal{F}}^{\Upsilon}$. We illustrate this fact for (IND-CPA).

Proposition 4.2. *Let $\mathsf{Ens}_{\mathrm{PPT}}^{\nu}$ be the category of ensembles of PPT-realized boolean functions modulo negligible functions. A crypto system in the usual sense (as described in the Introduction) is equivalent to an abstract crypto system in*

---

[5] Deterministic functions can be characterized intrinsically in $\mathsf{Sto}_{\mathcal{F}}$, $\mathsf{Ens}_{\mathcal{F}}$ and $\mathsf{Ens}_{\mathcal{F}}^{\Upsilon}$.

this category. Such a crypto system is semantically secure, i.e. it satisfies (IND-CPA), as defined by (2), if and only if the following diagram commutes for all arrows A_0 and A_1 in $\mathsf{Ens}_{\mathrm{PPT}}^{\nu}$.

$$
\begin{array}{ccccccc}
\mathcal{K} & \xrightarrow{\langle \mathrm{id}_{\mathcal{K}}, \mathsf{A}_0 \rangle} & \mathcal{K} \times \mathcal{M}^2 \times \mathcal{S} & \xrightarrow{\pi} & \mathcal{K} \times \mathcal{M} \times \mathcal{M}^2 \times \mathcal{S} & \xrightarrow{\mathsf{E} \times \mathrm{id}} & \mathcal{C} \times \mathcal{M}^2 \times \mathcal{S} \\
\big\downarrow{\scriptstyle !} & & & & & & \big\downarrow{\scriptstyle \mathsf{A}_1} \\
1 & & & \xrightarrow{\quad b \quad} & & & 2
\end{array}
$$

A similar proposition holds for (IND-CCA2).

5   Further work

While the various notions of secrecy can thus be characterized by commutative diagrams in suitable categories, the notions of one-way function and pseudo-random generator correspond to the requirements that some diagrams *do not* commute. This leads to interesting categorical structures, which seem to be best expressed in terms of enriched categories, and the suitable convolution operations. This observation led to an different approach, through *monoidal computer* [24,25], lifting the ideas from another strand of Lambek's work, leading from infinite abacus as an intensional model of computation [18], to the extensional models [20], elaborated in the book with P.J. Scott [21].

But how useful might our categorical models of computation be for cryptography? Can the categorical tools, developed for high level program semantics, really be used to stratify cryptographic constructions? The preliminary evidence, some of which was presented here, suggests that certain types of cryptographic proofs and constructions can be significantly simplified by using categorical tools to 'hide the implementation details'. The price to be paid, though, is that this hiding requires some preliminary work. For instance, we have seen that the secrecy conditions can be captured by simple diagrams, albeit in randomized categories. This approach echoes the well established programming methodologies, where complex structures are encapsulate into components that hide the irrelevant implementation details, and only the fragments that need to be manipulated are displayed at the interface. The categorical approach developed in Lambek's work has made such strategies available across a broad gamut of sciences.

References

1. Kim-Kwang Raymond Choo, Colin Boyd, and Yvonne Hitchcock. Errors in computational complexity proofs for protocols. In Bimal K. Roy, editor, *ASIACRYPT*, volume 3788 of *Lecture Notes in Computer Science*, pages 624–643. Springer, 2005.

2. Alexander W Dent. Fundamental problems in provable security and cryptography. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 364(1849):3215–3230, 2006.

3. Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

4. Danny Dolev, Shimon Even, and Richard M. Karp. On the security of ping-pong protocols. In *CRYPTO*, pages 177–186, 1982.

5. Danny Dolev and Andrew C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983.

6. Dusko Pavlovic. Geometry of abstraction in quantum computation. *Proceedings of Symposia in Applied Mathematics*, 71:233–267, 2012. arxiv.org:1006.1010.

7. Peter Freyd. *Abelian Categories: an Introduction to the Theory of Functors*. Harper and Row, 1964.

8. Oded Goldreich. *Foundations of Cryptography*. Cambridge University Press, 2000.

9. Shafi Goldwasser and Silvio Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. In *STOC '82: Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 365–377, New York, NY, USA, 1982. ACM Press.

10. Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

11. Pierre Antoine Grillet. *Semigroups: an introduction to the structure theory*. Marcel Dekker, Inc., 1995.

12. André Joyal and Ross Street. The geometry of tensor calculus I. *Adv. in Math.*, 88:55–113, 1991.

13. Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman & Hall/CRC Series in Cryptography and Network Security. Chapman & Hall/CRC, 2007.

14. Gregory Max Kelly. On clubs and doctrines. In Gregory Max Kelly, editor, *Category Seminar. Sydney 1972/73*, pages 181–256. Springer-Verlag, Berlin, 1974.

15. Neal Koblitz and Alfred Menezes. Another look at "Provable Security". II. In Rana Barua and Tanja Lange, editors, *INDOCRYPT*, volume 4329 of *Lecture Notes in Computer Science*, pages 148–175. Springer, 2006.

16. Neal Koblitz and Alfred Menezes. Another look at "Provable Security". *J. Cryptology*, 20(1):3–37, 2007.

17. Neal Koblitz and Alfred Menezes. The brave new world of bodacious assumptions in cryptography. *Notices of the American Mathematical Society*, 57(3):357–365, March 2010.

18. Joachim Lambek. How to program an infinite abacus. *Canad. Math. Bull.*, 4(3):295–302, 1961.

19. Joachim Lambek. *Lectures on Rings and Modules*. Blaisdell Publishing Co., 1966.

20. Joachim Lambek. From types to sets. *Adv. in Math.*, 36:113–164, 1980.

21. Joachim Lambek and Philip J. Scott. *Introduction to higher order categorical logic*, volume 7 of *Cambridge Stud. Adv. Math.* Cambridge University Press, New York, NY, USA, 1986.

22. Saunders Mac Lane. *Homology*. Springer-Verlag, 1963.

23. Dusko Pavlovic. Maps II: Chasing diagrams in categorical proof theory. *J. of the IGPL*, 4(2):1–36, 1996.

24. Dusko Pavlovic. Categorical logic of names and abstraction in action calculus. *Math. Structures in Comp. Sci.*, 7:619–637, 1997.

25. Dusko Pavlovic. Monoidal computer I: Basic computability by string diagrams. *Information and Computation*, 2013. to appear; arxiv:1208.5205.

26. Dusko Pavlovic and Catherine Meadows. Bayesian authentication: Quantifying security of the Hancke-Kuhn protocol. *E. Notes in Theor. Comp. Sci.*, 265:97 – 122, 2010.

27. Dusko Pavlovic and Vaughan Pratt. The continuum as a final coalgebra. *Theor. Comp. Sci.*, 280(1–2):105–122, 2002.

28. Charles Rackoff and Daniel R. Simon. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In Joan Feigenbaum, editor, *CRYPTO*, volume 576 of *Lecture Notes in Computer Science*, pages 433–444. Springer, 1991.

29. Claude E. Shannon. Communication theory of secrecy systems. *Bell Systems Technical Journal*, 28:656–715, 1949.

30. Victor Shoup. OAEP reconsidered. In *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '01, pages 239–259, London, UK, 2001. Springer-Verlag.