1.4. Motivation to the Rigorous Treatment

In this section we address three related issues:

1. the mere need for a rigorous treatment of the field,
2. the practical meaning and/or consequences of the rigorous treatment, and
3. the "conservative" tendencies of the treatment.

Parts of this section (corresponding to Items 2 and 3) are likely to become more clear after reading any of the following chapters.

1.4.1. The Need for a Rigorous Treatment

If the truth of a proposition does not follow
from the fact that it is self-evident to us,
then its self-evidence in no way justifies our belief in its truth.
Ludwig Wittgenstein, *Tractatus logico-philosophicus* (1921)

Cryptography is concerned with the construction of schemes that will be robust against malicious attempts to make these schemes deviate from their prescribed functionality. Given a desired functionality, a cryptographer should design a scheme that not only will satisfy the desired functionality under "normal operation" but also will maintain that functionality in face of adversarial attempts that will be devised after the cryptographer has completed the design. The fact that an adversary will devise its attack after the scheme has been specified makes the design of such schemes very hard. In particular, the adversary will try to take actions other than the ones the designer has envisioned. Thus, *the evaluation of cryptographic schemes* must take account of a practically infinite set of adversarial strategies. It is useless to make assumptions regarding the specific *strategy* that an adversary may use. The only assumptions that can be justified will concern the computational *abilities* of the adversary. To summarize, an evaluation of a cryptographic scheme is a study of an infinite set of potential strategies (which are not explicitly given). Such a highly complex study cannot be carried out properly without great care (i.e., rigor).

The design of cryptographic systems must be based on *firm foundations*, whereas ad hoc approaches and heuristics are a very dangerous way to go. Although always inferior to a rigorously analyzed solution, a heuristic may make sense when the designer has a very good idea about the environment in which a scheme is to operate. Yet a cryptographic scheme has to operate in a maliciously selected environment that typically will transcend the designer's view. Under such circumstances, heuristics make little sense (if at all).

In addition to these straightforward considerations, we wish to stress two additional aspects.

On Trusting Unsound Intuitions. We believe that *one* of the roles of science is to formulate, examine, and refine our intuition about reality. A rigorous formulation is

required in order to allow a careful examination that may lead either to verification and justification of our intuition or to its rejection as false (or as something that is true only in certain cases or only under certain refinements). There are many cases in which our initial intuition turns out to be correct, as well as many cases in which our initial intuition turns out to be wrong. The more we understand the discipline, the better our intuition becomes.

At this stage in history, it would be very presumptuous to claim that we have good intuition about the *nature of efficient computation*. In particular, we do not even know the answers to such basic questions as whether or not $\mathcal{P}$ is strictly contained in $\mathcal{NP}$, let alone have an understanding of what makes one computational problem hard while a seemingly related problem is easy. Consequently, we should be extremely careful when making assertions about what can or cannot be efficiently computed. Unfortunately, *making assertions about what can or cannot be efficiently computed is exactly what cryptography is all about*. Worse yet, many of the problems of cryptography have much more complex and cumbersome descriptions than are usually encountered in complexity theory. To summarize, cryptography deals with very complex computational notions and currently must do so without having a good understanding of much simpler computational notions. Hence, our current intuitions about cryptography must be considered highly unsound until they can be formalized and examined carefully. In other words, the general need to formalize and examine intuition becomes even more acute in a highly sensitive field such as cryptography that is intimately concerned with questions we hardly understand.

The Track Record. Cryptography, as a discipline, is well motivated. Consequently, cryptographic issues are being discussed by many researchers, engineers, and laypersons. Unfortunately, most such discussions are carried out without precise definitions of the subject matter. Instead, it is implicitly assumed that the basic concepts of cryptography (e.g., secure encryption) are self-evident (because they are so natural) and that there is no need to present adequate definitions. The fallacy of that assumption is demonstrated by the abandon of papers (not to mention private discussions) that derive and/or jump to wrong conclusions concerning security. In most cases these wrong conclusions can be traced back to implicit misconceptions regarding security that could not have escaped the eyes of the authors if they had been made explicit. We avoid listing all such cases here for several obvious reasons. Nevertheless, we shall mention one well-known example.

Around 1979, Ron Rivest claimed that no signature scheme that was "proven secure assuming the intractability of factoring" could resist a "chosen message attack." His argument was based on an implicit (and unjustified) assumption concerning the nature of a "proof of security (which assumes the intractability of factoring)." Consequently, for several years it was believed that one had to choose between having a signature scheme "proven to be unforgeable under the intractability of factoring" and having a signature scheme that could resist a "chosen message attack." However, in 1984, Goldwasser, Micali, and Rivest pointed out the fallacy on which Rivest's 1979 argument had been based and furthermore presented signature schemes that could resist a "chosen message attack," under general assumptions. In particular, the intractability of factoring suffices

to prove that there exists a signature scheme that can resist "forgery," even under a "chosen message attack."

To summarize, the basic concepts of cryptography are indeed very natural, but they are *not* self-evident nor well understood. Hence, we do not yet understand these concepts well enough to be able to discuss them *correctly* without using precise definitions and rigorously justifying every statement made.

1.4.2. Practical Consequences of the Rigorous Treatment

As customary in complexity theory, our treatment is presented in terms of asymptotic analysis of algorithms. (Actually, it would be more precise to use the term "functional analysis of running time.") This makes the treatment less cumbersome, but it is *not* essential to the underlying ideas. In particular, the definitional approach taken in this book (e.g., the definitions of one-way functions, pseudorandom generators, zero-knowledge proofs, secure encryption schemes, unforgeable signature schemes, and secure protocols) is based on general paradigms that remain valid in any reasonable computational model. In particular, the definitions, although stated in an "abstract manner," lend themselves to concrete interpolations. The same holds with respect to the results that typically relate several such definitions. To clarify the foregoing, we shall consider, as an example, the statement of a generic result as presented in this book.

A typical result presented in this book relates two computational problems. The first problem is a simple computational problem that is assumed to be intractable (e.g., intractability of factoring), whereas the second problem consists of "breaking" a specific implementation of a useful cryptographic primitive (e.g., a specific encryption scheme). The abstract statement may assert that if integer factoring cannot be performed in polynomial time, then the encryption scheme is secure in the sense that it cannot be "broken" in polynomial time. Typically, the statement is proved by a fixed polynomial-time reduction of integer factorization to the problem of breaking the encryption scheme. Hence, what is actually being proved is that if one can break the scheme in time $T(n)$, where n is the security parameter (e.g., key length), then one can factor integers of length m in time $T'(m) = f(m, T(g(m)))$, where f and g are fixed polynomials that are at least implicit in the proof. In order to determine the practicality of the result, one should first determine these polynomials (f and g). For most of the basic results presented in this book, these polynomials are reasonably small, in the sense that instantiating a scheme with a reasonable security parameter and making reasonable intractability assumptions (e.g., regarding factoring) will yield a scheme that it is infeasible to break in practice. (In the exceptional cases, we say so explicitly and view these results as merely claims of the plausibility of relating the two notions.) We actually distinguish three types of results:

1. *Plausibility results:* Here we refer to results that are aimed either at establishing a connection between two notions or at providing a generic way of solving a class of problems.

   A result of the first type says that, in principle, X (e.g., a specific tool) can be used in order to construct Y (e.g., a useful utility), but the specific construction provided in the proof may be impractical. Still, such a result may be useful in practice because it suggests that one may be able to use *specific* implementations of X in order to provide a

practical construction of Y. At the very least, such a result can be viewed as a challenge to the researchers to either provide a practical construction of Y using X or explain why a practical construction cannot be provided.

A result of the second type says that any task that belongs to some class $\mathcal{C}$ is solvable, but the generic construction provided in the proof may be impractical. Still, this is a very valuable piece of information: If we have a specific problem that falls into the foregoing class, then we know that the problem is solvable in principle. However, if we need to construct a real system, then we probably should construct a solution from scratch (rather than employing the preceding generic result).

To summarize, in both cases a plausibility result provides very useful information (even if it does not yield a practical solution). Furthermore, it is often the case that *some* tools developed toward proving a plausibility result may be useful in solving the specific problem at hand. This is typically the case for the next type of results.

2. *Introduction of paradigms and techniques that may be applicable in practice:* Here we refer to results that are aimed at introducing a new notion, model, tool, or technique. Such results (e.g., techniques) typically are applicable in practice, either as presented in the original work or, after further refinements, or at least as an inspiration.
3. *Presentation of schemes that are suitable for practical applications.*

Typically, it is quite easy to determine to which of the foregoing categories a specific result belongs. Unfortunately, the classification is not always stated in the original paper; however, typically it is evident from the construction. We stress that all results of which we are aware (in particular, all results mentioned in this book) come with an explicit construction. Furthermore, the security of the resulting construction is explicitly related to the complexity of certain intractable tasks. Contrary to some uninformed beliefs, for each of these results there is an explicit translation of concrete intractability assumptions (on which the scheme is based) into lower bounds on the amount of work required to violate the security of the resulting scheme.[9] We stress that this translation can be invoked for any value of the security parameter. Doing so will determine whether a specific construction is adequate for a specific application under specific reasonable intractability assumptions. In many cases the answer is in the affirmative, but in general this does depend on the specific construction, as well as on the specific value of the security parameter and on what it is reasonable to assume for this value (of the security parameter).

1.4.3. The Tendency to Be Conservative

When reaching the chapters in which cryptographic primitives are defined, the reader may notice that we are unrealistically "conservative" in our definitions of security. In other words, we are unrealistically liberal in our definition of insecurity. Technically speaking, this tendency raises no problems, because our primitives that are secure in a very strong sense certainly are also secure in the (more restricted) reasonable sense. Furthermore, we are able to implement such (strongly secure) primitives using

[9] The only exception to the latter statement is Levin's observation regarding the existence of a *universal one-way function* (see Section 2.4.1).

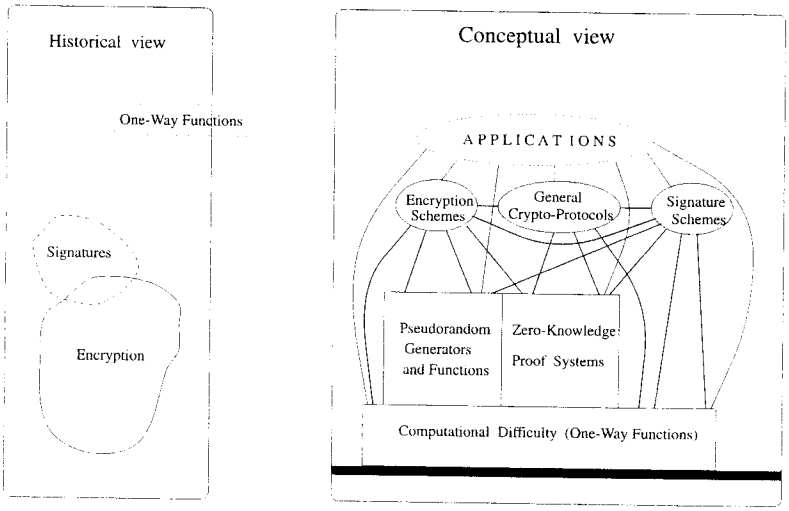

Figure 1.1: Cryptography: two points of view.

reasonable intractability assumptions, and in most cases we can show that such assumptions are necessary even for much weaker (and, in fact, less than minimal) notions of security. Yet the reader may wonder why we choose to present definitions that seem stronger than what is required in practice.

The reason for our tendency to be conservative when defining security is that it is extremely difficult to capture what is *exactly* required in a specific practical application. Furthermore, each practical application has different requirements, and it is undesirable to redesign the system for each new application. Thus, we actually need to address the security concerns of all future applications (which are unknown to us), not merely the security concerns of some known applications. It seems impossible to cover whatever can be required in all applications (or even in some wide set of applications) without taking our conservative approach.[10] In the sequel, we shall see how our conservative approach leads to definitions of security that can cover all possible practical applications.

1.5. Miscellaneous

In Figure 1.1 we confront the "historical view" of cryptography (i.e., the view of the field in the mid-1970s) with the approach advocated in this text.

1.5.1. Historical Notes

Work done during the 1980s plays a dominant role in our exposition. That work, in turn, had been tremendously influenced by previous work, but those early influences are not stated explicitly in the historical notes to subsequent chapters. In this section we shall

[10] One may even argue that it seems impossible to cover whatever is required in one reasonable application without taking our conservative approach.