Foundations of Cryptography – A Primer
Foundations of Cryptography
– A Primer

Oded Goldreich

Department of Computer Science
Weizmann Institute of Science
Rehovot Israel
oded.goldreich@weizmann.ac.il

now
the essence of knowledge
Boston – Delft
Contents

1 Introduction and Preliminaries 1
  1.1 Introduction 1
  1.2 Preliminaries 7

I Basic Tools 10

2 Computational Difficulty and One-way Functions 13
  2.1 One-way functions 14
  2.2 Hard-core predicates 18

3 Pseudorandomness 23
  3.1 Computational indistinguishability 24
  3.2 Pseudorandom generators 26
  3.3 Pseudorandom functions 29

4 Zero-Knowledge 33
Introduction and Preliminaries

It is possible to build a cabin with no foundations,
but not a lasting building.


1.1 Introduction

The vast expansion and rigorous treatment of cryptography is one of the major achievements of theoretical computer science. In particular, concepts such as computational indistinguishability, pseudorandomness and zero-knowledge interactive proofs were introduced, classical notions such as secure encryption and unforgeable signatures were placed on sound grounds, and new (unexpected) directions and connections were uncovered. Indeed, modern cryptography is strongly linked to complexity theory (in contrast to “classical” cryptography which is strongly related to information theory).

Modern cryptography is concerned with the construction of information systems that are robust against malicious attempts to make these systems deviate from their prescribed functionality. The prescribed functionality may be the private and authenticated communi-
cation of information through the Internet, the holding of tamper-proof
and secret electronic voting, or conducting any “fault-resilient” multi-
party computation. Indeed, the scope of modern cryptography is very
broad, and it stands in contrast to “classical” cryptography (which has
focused on the single problem of enabling secret communication over
insecure communication media).

The design of cryptographic systems is a very difficult task. One
cannot rely on intuitions regarding the “typical” state of the environ-
ment in which the system operates. For sure, the adversary attacking
the system will try to manipulate the environment into “untypical” states.
Nor can one be content with counter-measures designed to withstand
specific attacks, since the adversary (which acts after the design of the
system is completed) will try to attack the schemes in ways that are
different from the ones the designer had envisioned. The validity of the
above assertions seems self-evident, but still some people hope that in
practice ignoring these tautologies will not result in actual damage.
Experience shows that these hopes rarely come true; cryptographic
schemes based on make-believe are broken, typically sooner than later.

In view of the foregoing, we believe that it makes little sense to make
assumptions regarding the specific strategy that the adversary may use.
The only assumptions that can be justified refer to the computational
abilities of the adversary. Furthermore, the design of cryptographic sys-
tems has to be based on firm foundations; whereas ad-hoc approaches
and heuristics are a very dangerous way to go. A heuristic may make
sense when the designer has a very good idea regarding the environ-
ment in which a scheme is to operate, yet a cryptographic scheme has
to operate in a maliciously selected environment which typically tran-
scends the designer’s view.

This primer is aimed at presenting the foundations for cryptography.
The foundations of cryptography are the paradigms, approaches and
techniques used to conceptualize, define and provide solutions to nat-
ural “security concerns”. We will present some of these paradigms,
approaches and techniques as well as some of the fundamental results
obtained using them. Our emphasis is on the clarification of funda-
mental concepts and on demonstrating the feasibility of solving several
central cryptographic problems.
1.1. Introduction

Solving a cryptographic problem (or addressing a security concern) is a two-stage process consisting of a *definitional stage* and a *constructual stage*. First, in the definitional stage, the functionality underlying the natural concern is to be identified, and an adequate cryptographic problem has to be defined. Trying to list all undesired situations is infeasible and prone to error. Instead, one should define the functionality in terms of operation in an imaginary ideal model, and require a candidate solution to emulate this operation in the real, clearly defined, model (which specifies the adversary’s abilities). Once the definitional stage is completed, one proceeds to construct a system that satisfies the definition. Such a construction may use some simpler tools, and its security is proved relying on the features of these tools. In practice, of course, such a scheme may need to satisfy also some *specific* efficiency requirements.

This primer focuses on several archetypical cryptographic problems (e.g., encryption and signature schemes) and on several central tools (e.g., computational difficulty, pseudorandomness, and zero-knowledge proofs). For each of these problems (resp., tools), we start by presenting the natural concern underlying it (resp., its intuitive objective), then define the problem (resp., tool), and finally demonstrate that the problem may be solved (resp., the tool can be constructed). In the latter step, our focus is on demonstrating the feasibility of solving the problem, not on providing a practical solution. As a secondary concern, we typically discuss the level of practicality (or impracticality) of the given (or known) solution.

**Computational difficulty**

The aforementioned tools and applications (e.g., secure encryption) exist only if some sort of computational hardness exists. Specifically, all these problems and tools require (either explicitly or implicitly) the ability to generate instances of hard problems. Such ability is captured in the definition of one-way functions. Thus, one-way functions are the very minimum needed for doing most natural tasks of cryptography. (It turns out, as we shall see, that this necessary condition is “essentially” sufficient; that is, the existence of one-way functions (or augmentations
and extensions of this assumption) suffices for doing most of cryptography.)

Our current state of understanding of efficient computation does not allow us to prove that one-way functions exist. In particular, if $P = NP$ then no one-way functions exist. Furthermore, the existence of one-way functions implies that $NP$ is not contained in $BPP \supseteq P$ (not even “on the average”). Thus, proving that one-way functions exist is not easier than proving that $P \neq NP$; in fact, the former task seems significantly harder than the latter. Hence, we have no choice (at this stage of history) but to assume that one-way functions exist. As justification to this assumption we may only offer the combined beliefs of hundreds (or thousands) of researchers. Furthermore, these beliefs concern a simply stated assumption, and their validity follows from several widely believed conjectures which are central to various fields (e.g., the conjectured intractability of integer factorization is central to computational number theory).

Since we need assumptions anyhow, why not just assume what we want (i.e., the existence of a solution to some natural cryptographic problem)? Well, first we need to know what we want: as stated above, we must first clarify what exactly we want; that is, go through the typically complex definitional stage. But once this stage is completed, can we just assume that the definition derived can be met? Not really: once a definition is derived, how can we know that it can be met at all? The way to demonstrate that a definition is viable (and that the corresponding intuitive security concern can be satisfied at all) is to construct a solution based on a better understood assumption (i.e., one that is more common and widely believed). For example, looking at the definition of zero-knowledge proofs, it is not a-priori clear that such proofs exist at all (in a non-trivial sense). The non-triviality of the notion was first demonstrated by presenting a zero-knowledge proof system for statements, regarding Quadratic Residuosity, which are believed to be hard to verify (without extra information). Furthermore, contrary to prior beliefs, it was later shown that the existence of one-way functions implies that any NP-statement can be proved in zero-knowledge. Thus, facts that were not known to hold at all (and even believed to be false), were shown to hold by reduction to widely
believed assumptions (without which most of modern cryptography collapses anyhow). To summarize, not all assumptions are equal, and so reducing a complex, new and doubtful assumption to a widely-believed simple (or even merely simpler) assumption is of great value. Furthermore, reducing the solution of a new task to the assumed security of a well-known primitive typically means providing a construction that, using the known primitive, solves the new task. This means that we do not only know (or assume) that the new task is solvable but we also have a solution based on a primitive that, being well-known, typically has several candidate implementations.

**Prerequisites and structure**

Our aim is to present the basic concepts, techniques and results in cryptography. As stated above, our emphasis is on the clarification of fundamental concepts and the relationship among them. This is done in a way independent of the particularities of some popular number theoretic examples. These particular examples played a central role in the development of the field and still offer the most practical implementations of all cryptographic primitives, but this does not mean that the presentation has to be linked to them. On the contrary, we believe that concepts are best clarified when presented at an abstract level, decoupled from specific implementations. Thus, the most relevant background for this primer is provided by basic knowledge of algorithms (including randomized ones), computability and elementary probability theory.

The primer is organized in two main parts, which are preceded by preliminaries (regarding efficient and feasible computations). The two parts are **Part I – Basic Tools** and **Part II – Basic Applications**. The basic tools consist of computational difficulty (one-way functions), pseudo-randomness and zero-knowledge proofs. These basic tools are used for the basic applications, which in turn consist of Encryption Schemes, Signature Schemes, and General Cryptographic Protocols.

In order to give some feeling of the flavor of the area, we have included in this primer a few proof sketches, which some readers may find too terse. We stress that following these proof sketches is **not**
essential to understanding the rest of the material. In general, later sections may refer to definitions and results in prior sections, but not to the constructions and proofs that support these results. It may be even possible to understand later sections without reading any prior section, but we believe that the order we chose should be preferred because it proceeds from the simplest notions to the most complex ones.

Suggestions for further reading

This primer is a brief summary of the author’s two-volume work on the subject (65; 67). Furthermore, Part I corresponds to (65), whereas Part II corresponds to (67). Needless to say, the reader is referred to these textbooks for further detail.

Two of the topics reviewed by this primer are zero-knowledge proofs (which are probabilistic) and pseudorandom generators (and functions). A wider perspective on probabilistic proof systems and pseudorandomness is provided in (62, Sections 2–3).

Current research on the foundations of cryptography appears in general computer science conferences (e.g., FOCS and STOC), in cryptography conferences (e.g., Crypto and EuroCrypt) as well as in the newly established Theory of Cryptography Conference (TCC).
1.2. Preliminaries

Practice. The aim of this primer is to introduce the reader to the theoretical foundations of cryptography. As argued above, such foundations are necessary for sound practice of cryptography. Indeed, practice requires more than theoretical foundations, whereas the current primer makes no attempt to provide anything beyond the latter. However, given a sound foundation, one can learn and evaluate various practical suggestions that appear elsewhere (e.g., in [97]). On the other hand, lack of sound foundations results in inability to critically evaluate practical suggestions, which in turn leads to unsound decisions. Nothing could be more harmful to the design of schemes that need to withstand adversarial attacks than misconceptions about such attacks.

Non-cryptographic references: Some “non-cryptographic” works were referenced for sake of wider perspective. Examples include [4; 5; 6; 7; 55; 69; 78; 96; 118].

1.2 Preliminaries

Modern cryptography, as surveyed here, is concerned with the construction of efficient schemes for which it is infeasible to violate the security feature. Thus, we need a notion of efficient computations as well as a notion of infeasible ones. The computations of the legitimate users of the scheme ought be efficient, whereas violating the security features (by an adversary) ought to be infeasible. We stress that we do not identify feasible computations with efficient ones, but rather view the former notion as potentially more liberal.

Efficient computations and infeasible ones

Efficient computations are commonly modeled by computations that are polynomial-time in the security parameter. The polynomial bounding the running-time of the legitimate user’s strategy is fixed and typically explicit (and small). Indeed, our aim is to have a notion of efficiency that is as strict as possible (or, equivalently, develop strategies that are as efficient as possible). Here (i.e., when referring to the complexity of the legitimate users) we are in the same situation as in any algorithmic setting. Things are different when referring to our assumptions.
regarding the computational resources of the adversary, where we refer to the notion of feasible that we wish to be as wide as possible. A common approach is to postulate that feasible computations are polynomial-time too, but here the polynomial is not a-priori specified (and is to be thought of as arbitrarily large). In other words, the adversary is restricted to the class of polynomial-time computations and anything beyond this is considered to be infeasible.

Although many definitions explicitly refer to the convention of associating feasible computations with polynomial-time ones, this convention is inessential to any of the results known in the area. In all cases, a more general statement can be made by referring to a general notion of feasibility, which should be preserved under standard algorithmic composition, yielding theories that refer to adversaries of running-time bounded by any specific super-polynomial function (or class of functions). Still, for sake of concreteness and clarity, we shall use the former convention in our formal definitions (but our motivational discussions will refer to an unspecified notion of feasibility that covers at least efficient computations).

Randomized (or probabilistic) computations

Randomized computations play a central role in cryptography. One fundamental reason for this fact is that randomness is essential for the existence (or rather the generation) of secrets. Thus, we must allow the legitimate users to employ randomized computations, and certainly (since randomization is feasible) we must consider also adversaries that employ randomized computations. This brings up the issue of success probability: typically, we require that legitimate users succeed (in fulfilling their legitimate goals) with probability 1 (or negligibly close to this), whereas adversaries succeed (in violating the security features) with negligible probability. Thus, the notion of a negligible probability plays an important role in our exposition. One requirement of the definition of negligible probability is to provide a robust notion of rareness: A rare event should occur rarely even if we repeat the experiment for a feasible number of times. That is, in case we consider any polynomial-time computation to be feasible, a function \( \mu: \mathbb{N} \to \mathbb{N} \) is called negligible
if \(1 - (1 - \mu(n))^{p(n)} < 0.01\) for every polynomial \(p\) and sufficiently big \(n\) (i.e., \(\mu\) is negligible if for every positive polynomial \(p'\) the function \(\mu(\cdot)\) is upper-bounded by \(1/p'(\cdot)\)). However, if we consider the function \(T(n)\) to provide our notion of infeasible computation then functions bounded above by \(1/T(n)\) are considered negligible (in \(n\)).

We will also refer to the notion of noticeable probability. Here the requirement is that events that occur with noticeable probability, will occur almost surely (i.e., except with negligible probability) if we repeat the experiment for a polynomial number of times. Thus, a function \(\nu : \mathbb{N} \to \mathbb{N}\) is called noticeable if for some positive polynomial \(p'\) the function \(\nu(\cdot)\) is lower-bounded by \(1/p'(\cdot)\).
References


References


References


References

References


References


References


